January/February 1996 *Vol. 3 Number 4* *$4.50 Canada, $4.00 US*

the world of 68' micros

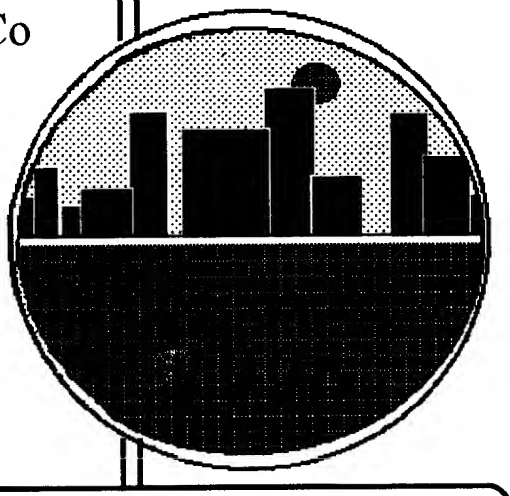
Supporting Tandy Color Computer Disk BASIC, CoCo OS-9, and OS-9/68000

WANTED:

Basic09 (and Microware Basic) programmer. Must be willing to convert a large Disk BASIC program to Basic09/MWBasic code and explain process line by line. Rewards include a few free issues of this magazine, getting printed, and earning the admiration and respect of fellow OS-9 and CoCo enthusiasts. See page 4 for details.

New! Lower subscription rates!
US/Mexico: $24/year
Canada: $30/year
See page 19 for
new lower rates on microdisk!!!

CONTENTS



This is the place to be in 1996, we hope to be there... will YOU??

POSTMASTER:
If undeliverable return to:
FARNA *Systems* PB
Box 321
Warner Robins, GA 31099

Address Correction Requested

the world of 68' micros

Published by:
FARNA Systems
P.O. Box 321
Warner Robins, GA 31099-0321

Editor: F. G. Swygert

Subscriptions:
$24/year (6 issues) US/Mexico; $30/year for Canada ($13 US/M, $16 Canada for six months- three issues). Overseas $50/year ($26 for three issues) AIR; $38/year, $18 six months for surface mail.
microdisk: $20/ year (3 issues) or $8 each US/Mexico, $25/year, $10 single for Canada, and $40/year, $15 single overseas airmail. Contains programs and source listings from magazine; not stand-alone.

Advertising Rates:
Contact publisher for rates. We have scales to suit every type business. Special rates for entreprenuers and first time advertisers.

Any and all contributions welcomed. Submission constitutes warranty on part of the author that the work is original and not copywritten by another party. All opinions expressed herein are those of the individual writers, not necessarily the publisher or editor. FARNA Systems reserves the right to edit or reject any submitted material without explanation. Renumeration discussed on an individual basis.

Back issues are $4 per copy. Overseas add $1 each for surface, $2.50 airmail delivery.

Newsstand/bulk orders available. Dealers should contact the publisher for details.

Problems with delivery, change of address, subscriptions, or advertisers should be sent to the publisher with a short description.

The publisher is available for comment via **e-mail** at dsrtfox@Delphi.com. The Delphi CoCo and OS-9 SIGs on Delphi are also frequented (The Delphi SIGs are still sponsored by **Falsoft**).

The editor speaks...

F.G. Swygert

Subscription renewals are very important for continued support. Many of you didn't like seeing the magazine suddenly drop to six issues per year. And I know readers probably aren't pleased with the size of the last two issues, nor this one. Do these two items spell the beginning of the end for support?

The minimum number of items that can be mailed third class is 200 each mailing. Although there are currently just under 250 subscribers, fifty are in Canada or other countries. These don't apply to the third class minimum. This leaves UNDER 200 for the continental U.S. My third class mailing permit is also due for renewal ($85). This issue was mailed first class in a slightly different format (no "hard" cover, folded, and stapled only in the corner... most of you save them in binders anyway) because of this. If I don't get a lot of renewals or new subscriptions, there won't be an average of 200 issues sent in the continental states. So it doesn't make good sense to renew the third class mailing permit. The new format and an average of twenty pages keeps the magazine below weight requirements. Even then, mailing cost will more than double (from .255 each to .55 each). The smaller format will make up for some of the increased postage due to the reciprocal decrease in printing, of course.

The bottom line is that I would like to keep the existing format and keep the page count over 26. But this can only be done if renewals and new subscriptions come in and the US subscriber count stays above 200. This is YOUR magazine... it is up to each and every subscriber to keep it alive. And as long as there is adequate interest, it will live, even if it is a bit smaller.

You are not benefitting only yourself by renewing. You are also doing all the others who continue to use their CoCos and 68K computers a favor... you are helping to keep their systems alive!

How do you tell if it is time to renew? Take a good look at your mailing label. Beside your name is a month and year (MM/YY). That is the month you should receive your last issue. I will write "LAST ISSUE" on your label on that month. Usually, one issue past that month will be sent, again with "LAST ISSUE" on the label, just to keep anyone from accidentally missing an issue, but this time I mean it! So don't delay renewing to long, or you WILL miss it! **I'll end my renewal tirade with this: rest assured that we will continue publication at least through 1996.** We'll see about 1997 as the year continues, but I'll know when the big renewal "season" comes in August. **Bottom line: keep renewing and I'll keep publishing!**

On an upbeat note, I do have some ideas to increase circulation. The best, I think, is to add some more microcontroller support. The 68HC11 uses a core processor very similar to the 6809, and the 68300 series use 68000 cores. In fact, if you can program m/l code on a CoCo, you can program a 68HC11. Since the 68300 code is identical to 68000, the same is obviously true there. So expect to see some articles on these in the future.

One more item of business before I close... One subscriber asked if they would be getting eight issues anyway after the change to bi-monthly publication, their reasoning being they had paid for eight issues. The simple answer is NO... subscription are by year, not by issue. The reality of the situation is that this presents a problem, since the months fall differently. So the real answer is that those who subscribed or renewed before the change will actually receive an "extra" issue this year, for a total of seven (instead of six or eight), just because that's the best way to work the number of issues and renewal dates out, and to be fair to everyone.

Note that the prices for microdisk has changed to $20/year US ($25 Canada, $40 overseas), published one disk for every two magazine issues, or $8 each issue US ($10.00 US for Canada, $15.00 US overseas). Previous subscribers and all those who renewed at the old rates will receive two extra issues.

Please don't be disheartened by the size and appearance of this issue. I actually think the content is very good though lean... there is lots of meat here and little fat. That is what I expect to do with coming issues also... to continue running very good, informative articles that are useful to the reader.

In some ways, that makes each issue a little harder to put together – squeezing the best into the available space after gleaning through a lot of information. In another, it makes for easier filling of an issue. This is not due to less space to fill, but more to the lessened need to find short articles to fit the empty spaces.

I think you'll find the useful content of this issue to be of a very good quality. And I hope you continue your subscriptions and support even though we are a little leaner. We are far from finished... until YOUR support ends...

Fran G. Swygert

< 268'm >

Letters to the Editor

This is in response to the letter from Roger Dedon in the November/December issue of 268'm. Overall, all is indeed well with the OS-9 Users group. Roger has received all the issues of the newsletter printed since his membership was received by the Users Group. His membership (mailed by Roger 30 December 1994) arrived after the first issue of 1995 had been sent out, and the second issue was mailed out about the time we received Roger's query (22 March 1995.)

The newsletter is published four to six times per year (as set in our bylaws), depending on available material. Earlier incarnations of the Users Group had no minimum requirement that I am aware of. In 1994, four issues were printed, totalling (including covers) 56 pages, plus a special edition distributed at the Chicago Fest with 14 pages. In 1995 (to date) four issues have been printed, totalling 98 pages. A fifth issue for 1995 is being put together as I write this, and should be in the mail by mid-December.

Material covered is all OS-9 related, covering OS-9/6809, OS-968000, and OS-9/9000, along with both personal use and industrial material.

Any questions regarding the UG can be sent to the address below, or to our Internet address (os9ug@sandv.chi.il.us)

OS-9 Users Group
Suite 109
6158 W 63rd St.
Chicago, IL 60638

I see from past issues you now sell the Ken-Ton hard disk system. I purchased one of these units several years ago. My problem is that either the drive or the power supply has gone bad. I suspect the drive. The system boots but after some time the drive stops spinning. While I have not lost any data as yet the system is totally unreliable. Do you know of a source for a replacement drive? Everything I find in magazines are IDE or SCSI2 and much larger than my needs or system capacity.

John M. Schuster
8800 Junipero Ave.
Atascadero, CA 93422

John, the best source I have found for inexpensive, small (under 500MB) SCSI drives is Hi-Tech (59 South La Patera Lane, Goleta, CA 93117; 805-681-9961; e-mail through their web page: http//www.internet-cafe.com/hi-tech/). They typically have drives in the 50-120MB range for under $100. Do note that these are usually refurbished drives, but I have bought several from them with no problems. They offer a 90 day warranty on all drives, more if new. Note that SCSI2 drives work fine with SCSI1 (original SCSI) controllers.

After a great deal of thought (and a little procrastination,) I have decided to renew for another year. However, this will probably be the last renewal, even if you decide to continue publishing next year. I have pretty much moved from my CoCo 3 to my PC clone over the last few years. I still have the CoCo set up, but about the only things I do on it consistently is play Shanghai.

Even though I don't do much with my CoCo anymore, I'd like to wish you good luck in continuing to publish two68m. The first two years were generally interesting, and good quality considering the limited readership and dearth of authors with good submissions. I wouldn't be at all unhappy to see more by Robert Gault or Rick Ulland. I especially like hardware articles if you can get any (tho not real basic stuff.) I used to run OS-9 L2 on my CoCo3 but that doesn't interest me these days. Bob Gault's "Hi-lights from the past" articles are interesting and similar articles would be good.

Dick Batt

p.s. I'm glad no one was hurt during either of your car accidents, but I'd be especially vigilant if I were you as they say all things come in threes.

Thanks for the well wishes Dick! Unfortunately, you echo the sentiments of many CoCo users. I'm not so sure the emulators helped anything, except maybe to hurry the CoCo to the closet. Do rest assured that as long as there are enough readers to support the magazine it will continue to be published.

Thank you for publishing my calender program in the November/ December issue. I hope your readers will find it interesting and useful.

I am still trying to work out Multi-Vue. Interestingly, it appears to have been a forerunner of the current popular Windows 95. It would be nice if you or one of your knowledgeable readers could show novices such as me how to program Multi-Vue for our Color Computer 3's.

Best of luck to you in continuing your support of the Color Computer.

Burton Parke

Burton, if you look at some of Rick Ulland's past columns, you should find some hints on running Multi-Vue. The big news is that Rick takes an in-depth look at running and setting up Multi-Vue in this issue! Your letter was perfectly timed...

< 268'm >

INSIDE THE NEXT ISSUE:

Motorola 68K port of Linux
High Density Disk Hack for CoCo
And the usual stuff...

Checks 3

Malcolm Heath

A very good DECB checkbook program.

WANTED: Basic09 programmer...

For some time now, I have been keeping my checking accounts up to date with a neat program on my CoCo. The DECB program is a freeware program that I downloaded from Delphi. What I want to do is list that program for our DECB readers, then have someone go through it line by line and convert it as directly as possible to Basic09 and Microware Basic (my understanding is the B09 and MWB are virtually the same, just for different processors... 6809 vs. 680x0).

Before any of you OS-9 people object, I know this isn't the most efficient way to do this. The purpose of this excercise is NOT to create really efficient Basic09 code to show case the strengths of Basic09. The purpose is to first show that DECB programs CAN be converted, and to give specific, line by line examples of how to emulate each command or get exactly the same results. Doing it this way illustrates converting on a command by command basis, which will give examples that can be followed out of context of the specific program.

After this is done, the writer can go back and break the program back into segments and show how each can be streamlined using Basic09's added features and power. This gives us a two-fold objective: converting programs directly, then optomizing the code for greater speed and efficiency.

To start things off, here is the first portion of the Disk BASIC code. Hopefully, someone will have volunteered to head the conversion project by the time the next issue is out with the remainder of the DECB code. If not, at least DECB people will have a good checkbook program!

CHECKS3 Program Notes

Showing the correct spacing in lines has always been a problem! After typing in the program, you'll have to RUN it then adjust spacing as necessary. Don't forget to make a save every time you change something, and keep at least one "old" save handy, just incase you make a big mistake.

Edit lines 7000-7060 with expense codes suitable to your needs. The codes are only used when printing a report by code, you need not use codes or can type any code when entering data. And of course one should edit the lines with the bank names as appropriate.

Personally, I keep one copy of the program on a disk with the data for each of my banks (I keep a personal and business account) rather than two accounts on one program. Makes keeping things on track easier!

```
10 'THIS PROGRAM PLACED IN THE
PUBLIC DOMAIN BY THE AUTHOR DEC
15,1987
20'**************************
30 '* NO HASSLE CHECK REGISTER*
40 '* A User Friendly Program *
50 '*    BY MALCOLM HEATH     *
60 '*    RYE, N.H. 03870      *
70'**************************
80 RGB:WIDTH80:CLS3:PALETTE2,56
90 POKE 65496,0:POKE 150,18:POKE
282,1
100 CLEAR500:DIM LC$(20),MC$(20)
110 Y$="L64;T5;V15;O5;D;D#;E;F;F#;G":
N$="L64;T5;V15;O4;G;F#;F;E;D#;D"
120 LC$=STRING$(79," "):MC$=STRING$
(29," ")
130 ON BRK GOTO 9010:VERIFYON
140 LOCATE0,0:ATTR2,1:FORZ=1TO3:
PRINTLC$(Z):NEXT
150 LOCATE2,0:PRINT"Version 1.3":
LOCATE26,1:PRINT"NO HASSLE CHECK
REGISTER":LOCATE62,0:PRINT"By
Malcolm Heath":LOCATE0,3:ATTR2,2
160 LOCATE0,22:ATTR2,7:PRINTLC$;
170 LOCATE17,22:PRINT"AMERICAN
FEDERAL BANK - FARNA SYSTEMS
ACCOUNT"
180 LOCATE30,17:ATTR2,2:PRINT"Press
Any Key":EXEC44539
190 FI$="NF"
200 WB$="AMERICAN FEDERAL BANK"
210 UNLOAD:LOCATE34,6:PRINT
"Selections"
220 LOCATE25,8:PRINT"1) CHECKBOOK
ENTRIES"
230 LOCATE25,9:PRINT"2) DISPLAY
ENTRIES"
240 LOCATE25,10:PRINT"3) REVIEW
EXPENSES"
250 LOCATE25,11:PRINT"4) BALANCE
THE CHECKBOOK"
260 LOCATE25,12:PRINT"5) CORRECT/
DELETE ENTRY"
280 LOCATE25,13:PRINT"6) QUIT"
290 LOCATE30,17:PRINT"<CHOOSE 1 -
6> ";
300 K$=INKEY$:IFK$=""THEN300
310 ON VAL(K$) GOTO1000,2000,3000,
4000,5000,8510: GOTO300
1000 '** CHECK MAINTENANCE
ROUTINE **
1010 GOSUB7500
1020 BAL=0:OBAL=0:AMT=0:LE=3:ENT=0
1030 FOR B=1 TO LOF(1)
1040 IF LOF(1)<1 THEN1080
1050 GET#1,B:BAL=BAL+CVN(AMT$)
1060 NEXTB
1070 OBAL=BAL
1080 CLS:ENT=LOF(1)
1090 GOSUB7000
1100 LOCATE0,23:ATTR2,0:PRINTLC$;
1110 LOCATE1,23:PRINT"LAST ENTRY -
> Date ";DA$;" - Chk # ";CH$;" - Amt.";
1120 IFEC$="DD"OREC$="II" THEN
PRINTUSING"$$##,###.##"CVN(AMT$);:
GOTO1140
1130 PRINTUSING"$$##,###.##";-
CVN(AMT$);
1140 PRINT" - BAL.";:PRINT USING
"$$##,###.##";OBAL;
1150 LOCATE0,0::ATTR2,1:PRINTLC$;:
LOCATE4,0:PRINTWB$;" CHECKS";
1160
LOCATE49,0:ATTR2,1:PRINT"Check
<";:ATTR7,4,B:PRINT"C";:ATTR2,1:PRINT">
Deposit <";:ATTR7,4,B:PRINT"D";
:ATTR2,1:PRINT"> Menu <"; :ATTR7,4,
B:PRINT"M";: ATTR2,1:PRINT">";
:ATTR2,2
1170 ENT=ENT+1
1180 K$=INKEY$:IFK$=""THEN1180
1190 IF K$="C" OR K$="D"THEN1200
ELSEIFK$="M"THEN9010 ELSE1180
1200 LOCATE49,0:ATTR7,4:PRINT"
PRESS BREAK TO ABORT
";:GOTO1210
1210 IF K$="C"THEN1260 ELSEIF
K$="D"THENE$="DD":GOTO1220ELSE1180
1220 LOCATE3,2:ATTR3,2:PRINT"Date
of Deposit":LOCATE25,2:PRINT"Amount
of Deposit":LOCATE66,2:PRINT"New Bal
No."
1230 ATTR2,2:LOCATE1,LE:LINEINPUT
D$:LOCATE23,LE:LINEINPUT"$";A$
1240 AMT=VAL(A$):BAL=BAL+AMT:
LOCATE62,LE:PRINTUSING"
$$##,###.##";BAL:LOCATE75,LE:PRINTENT;
1250 C$="":P$="      DEPOSIT
":GOTO1290
1260 LOCATE3,2:ATTR3,2:PRINT"Date
Check No.  Amount      Paid To       Exp.
Cat.  New Bal.  No."
1270 ATTR2,2:LOCATE1,LE:LINEINPUT
D$:LOCATE13,LE:LINEINPUTC$:LO
CATE23,LE:LINEINPUT"$":A$:LOCATE33,LE:
LINEINPUTP$:LOCATE57,LE:LINEINPUTE$
1280 AMT=VAL(A$):AMT=-
AMT:BAL=BAL+
AMT:LOCATE62,LE:PRINTUSING"
$$##,###.##";BAL:LOCATE75,LE:PRINTENT;
1290 LOCATE49,0:ATTR2,7:PRINT"
Correct <":ATTR2,7,B:PRINT"Y";:ATTR
2,7:PRINT"> or <";:ATTR2,7,B:PRINT"N";
:ATTR2,7:PRINT">      ";:ATTR2,2
1300 K$=INKEY$:IFK$=""THEN1300
1310 IFK$="Y"THEN1320 ELSEIFK$="N"
THENLOCATE1,LE:PRINTLC$:ENT=ENT-
1:BAL=BAL-AMT:GOTO1160ELSE1300
1320 LSET DA$=D$:LSET CH$=C$:LSET
AMT$=MKN$(AMT):LSET PA$=P$:LSET
BN$=FI$:LSET EC$=E$:LSET CL$="N"
1330 PUT#1,ENT
1340 BAL=BAL:AMT=0:LE=LE+1
1350 IF LE=17 THENLOCATE0,3:FOR
Z=1 TO 13:PRINTLC$(Z):NEXT:LE=3
1360 GOTO1160
2000 '** DISPLAY ENTRIES AND
BALANCES
2010 GOSUB7500
2020 CLS:BAL=0:TL=0:AMT=0:LE=2:
WM$="": DV$=""
2030 ATTR2,5:LOCATE0,0:PRINTLC$;
2040 LOCATE5,0:PRINT"Display ALL
```

```
Entries or select a MONTH
?";:LOCATE60,0:PRINT"      <A> or <M>";
2050 A$=INKEY$:IF A$=""THEN2050
2060 IF A$="A" OR A$="M" THEN
2070ELSE2050
2070 IF A$="M" THENLOCATE50,0:
ATTR2,1:PRINTMC$;:LOCATE57,0:LINEINPUT
"What Month <MMM> ";WM$
2080 ATTR2,2:CLS:LOCATE18,10:PRINT
"Do you want a Printed Copy    <Y or N> ?
"
2090 K$=INKEY$:IFK$=""THEN2090
2100 IF K$="Y" THEN
DV$="P":GOTO2120
2110 IF K$="N" THEN2120ELSE2090
2120 LOCATE8,10:PRINTSTRING$(50,"
")
2130 LOCATE0,0:ATTR2,5:PRINT" Date
Check No. Amount        Paid To
Bank Exp. Balance  Clr NO.
";:ATTR2,2:LOCATE0,1
2140 IFDV$="P"THENGOSUB6000
2150 FORE=1 TO LOF(1)
2160 IF LOF(1)=0 THENGOTO8010
2170 GET#1,E:BAL=BAL+CVN(AMT$)
2180 IF A$= "M" AND
LEFT$(DA$,3)<>WM$ THENLE=LE-
1:GOTO2260
2190
LOCATE1,LE:PRINTDA$:LOCATE12,
LE:PRINTCH$:LOCATE17,LE
2200 IF EC$="DD"OREC$="II" THEN
PRINT
USING"$$##,###.##";CVN(AMT$):
LOCATE 31, LE:ATTR2,1:PRINTPA$;:
LOCATE51,LE:ATTR2,2:GOTO2220
2210 PRINTUSING"$$##,###.##";-CVN
(AMT$):LOCATE31,LE:PRINTPA$;
2220 LOCATE53,LE:PRINTBN$:LOCATE
57,LE:PRINTEC$:LOCATE60,LE:PRINT
USING"$$# #,###.##";BAL:LOCATE73,
LE:PRINTCL$:LOCATE75,LE:PRINTE;
2230 IFDV$="P" THENGOSUB6080
2240 TL=TL+1:GOTO2250
2250 IF LE=22 THEN LOCATE 0,LE+1:
ATTR2,7:PRINT"      Listing Continues at
Top of Screen       Press any Key
":EXEC44539:LOCATE0,LE+1:ATTR2,2:
PRINTLC$;:LE=1
2260 LE=LE+1:NEXTE
2270 IF TL=0 THEN8010 ELSE2280
2280 IF A$="M" THEN LOCATE0,LE:
ATTR2,0:
PRINTLC$;:LOCATE5,LE:PRINT
"CHECKBOOK BALANCE IS";:PRINT
USING"$$###,###.##";BAL;:PRINT
"Press any Key for Menu";
2290 IF A$="A"THEN LOCATE0,LE:
ATTR2,0:PRINTLC$;:LOCATE7,LE:PRINT"This
is your Latest Entry    ^ ^ ^    Press Any
Key for MENU";
2300 IF DV$="P" THENGOSUB6120
2310 EXEC44539:GOTO9010
3000 '** EXPENSE LIST **
3010 GOSUB7500
3020 CLS:LE=2:BAL=0:AMT=0:TL=0:
PE$="": DV$=""
3030 GOSUB7000:LOCATE0,0:ATTR2,5:
PRINTLC$;
3040 LOCATE6,0:PRINT"Select Category
to Review ";:LOCATE 54,0:LINEINPUT"
Enter Expense Cat. ";E$:ATTR2,2
3050 LOCATE50,0:ATTR2,1:PRINTMC$;:
LOCATE51,0:LINEINPUT"MONTH or ALL
<MMM or A> ";PE$:ATTR2,2
3060 LOCATE0,18:FORZ=1TO5:PRINT
LC$(Z): NEXT
3070 LOCATE18,10:PRINT"Do you want
a Printed Copy    <Y or N> ?"
3080 K$=INKEY$:IFK$=""THEN3080
3090 IF K$="Y"THEN DV$="P"ELSEIF
K$="N"THEN3100ELSE3080
3100 LOCATE18,10:PRINTSTRING$(50,"
");
3110 LOCATE0,0:ATTR2,5:PRINT" Date
Check No. Amount        Paid To
Bank Exp.  Clr NO.
";:ATTR2,2:LOCATE0,1
3120 IF DV$="P"THENGOSUB6500
3130 FORS=1 TO LOF(1)
3140 IF LOF(1)=0THENGOTO8010
3150 GET#1,S
3160 IF E$=EC$ AND PE$=LEFT$(DA$,3)
THEN3180 ELSE3170
3170 IF E$=EC$ AND PE$="A"
THEN3180 ELSE3250
3180 LOCATE1,LE:PRINTDA$:LOCATE
12,LE: PRINTCH$:LOCATE17,LE
3190 IF EC$="DD"OREC$="II"THEN
PRINTUSING"$$##,###.##";CVN(AM
T$):LOCATE31,LE:ATTR2,1:PRINTPA$;:
LOCATE51,LE:ATTR2,2:GOTO3210
3200 PRINTUSING"$$##,###.##";-CVN
(AMT$);:LOCATE31,LE:PRINTPA$;
3210 LOCATE57,LE:PR INTBN$:LOCATE
63,LE:PRINTEC$:LOCATE70,LE:PRINT
CL$:LOCATE74,LE:PRINTS;:TL=TL+1
3220 IF DV$="P"THEN GOSUB6560
3230 BAL=BAL+CVN(AMT$)
3240 IF LE=22 THEN LOCATE 0,LE+1:
ATTR 2,6: PRINT"      Listing Continues
at top of Screen       Press Any Key
";:EXEC44539:
LOCATE0,LE+1:ATTR2,2:PRINTLC$;:LE=1
3250 IF E$<>EC$ OR
PE$<>LEFT$(DA$,3) THENLE=LE-1
3260 IF E$=EC$ AND PE$="A" THEN
LE=LE+1
3270 LE=LE+1:NEXTS
3280 IF TL=0 THEN8010 ELSE3290
3290 IF PE$="A" THEN3300 ELSE3310
3300 LOCATE0,LE:ATTR2,0: PRINTLC$;:
LOCATE20,LE:PRINT" Total for Category
";E$;" is";:GOTO3320
3310
LOCATE0,LE:ATTR2,0:PRINTLC$;LOCATE21,LE:PRINTPE$"
Total for Category ";E$;" is";:GOTO3320
3320 IFE$="DD"OR E$="II"
THENBAL=+BAL ELSEIF E$<>"DD"OR
E$<>"II"THEN BAL=-BAL
3330 PRINTUSING"$$###,###.##";BAL;
3340 IF DV$="P"THEN GOSUB6600
3350 LOCATE0,23:ATTR2,7:PRINTLC$;:
LOCATE 0, 23: PRINT"  Review Another
EXPENSE CATEGORY or Return to
MENU    Press <A> or <M> ";:ATTR2,2
3360 K$=INKEY$:IF K$=""THEN3360
3370 IF K$="A"THEN3380 ELSEIF
K$="M" THEN9010ELSE3360
3380 CLOSE#1:GOTO3000
4000 '** CHECKBOOK BALANCE
ROUTINE **
4010 B1=0:B2=0:B3=0:B4=0:C=0:D=0:
CR=0:FE=0: AMT=0:I N=0:BC=0
4020 GOSUB7500
4030
CLS:LOCATE0,0:ATTR2,0:PRINTLC$;
4040 LOCATE20,0:PRINT"Balancing
"WB$" Checks":ATTR2,2
4050 LOCATE4,4:PRINT"Date    Check
No.  Amount        Paid To        E.Cat.
Clr.  NO.";
4060 FOR M=1 TO LOF(1)
4070 IF LOF(1)=0THEN8010
4080 GET#1,M:IF CL$="N" THEN4090
ELSE4210
4090 LOCATE2,6:PRINTDA$:LOCA
TE13,6:
ATTR2,1:PRINTCH$;:ATTR2,2:LOCATE22,6:
ATTR2,1
4100 IFEC$="DD"THENPRINTUSING
"$$##,###.##"";CVN(AMT$);:ATTR2,2:
GOTO4120
4110 PRINTUSING"$$##,###.##";-
CVN(AMT$);:ATTR2,2
4120
LOCATE36,6:PRINTPA$:LOCATE61,6:PRINTEC$:
LOCATE69,6:PRINTCL$:LOCATE75,6:PRINTM;
4130 LOCATE24,12:PRINT"Has this Entry
Cleared <Y or N> "
4140 K$=INKEY$:IF K$="" THEN4140
4150 IF K$="Y" THEN
LOCATE0,6:PRINTLC$:V$="C":PLAYY$:GOTO4180
4160 IF K$="N" THEN
LOCATE0,6:PRINTLC$:PLAYN$:GOTO4210
4170 GOTO4140
4180 IF EC$="DD"
THENB1=B1+CVN(AMT$):D=D+1
4190 IF EC$<>"DD"
THENB2=B2+CVN(AMT$):C=C+1
4200 LSET CL$=V$:PUT#1,M
4210 AMT=0:NEXTM
4220 CLS:M=LOF(1):D$=""
4230 LOCATE20,12:PRINT"Did you
receive Interest this period ?"
4240 K$=INKEY$:IFK$=""THEN4240
4250 IF K$="Y"THEN CR=1:GOTO4270
4260 IF K$="N"THEN4320 ELSE4240
4270
CLS:M=M+1:LOCATE30,12:LINEINPUT"Todays
Date is: ";D$
4280 LOCATE24,12:LINEINPUT"Amount
of Intest Received:
$";IN$:AMT=VAL(IN$)
4290 LSET DA$=D$:LSET CH$="":LSET
AMT$=MKN$(AMT):LSET PA$="
INTEREST  ":LSET BN$=FI$:LSET
EC$="II":LSET CL$="C"
4300 PUT#1,M
4310 IN=AMT:AMT=0
4320 CLS:LOCATE30,12:PRINT"Any
Bank Charges ?"
4330 K$=INKEY$:IF K$=""THEN4330
4340 IF K$="Y"THEN FE=1:GOTO 4360
4350 IF K$="N"THEN4400ELSE4330
4360 IF D$="" THEN  LOCATE30,12:
LINEINPUT"Todays Date is:  ";D$:
GOTO4370
4370 M=M+1:LOCATE28,12:LINEINPUT"
Enter Bank Charge:   $";BC$:
AMT=VAL(BC$):AMT=-AMT
4380 LSET DA$=D$:LSET CH$="":LSET
AMT$=MKN$(AMT):LSET PA$="    BANK
CHARGE  ":LSET BN$=FI$:LSET
EC$="BC":LSET CL$="C":PUT#1,M
4390 BC=AMT:AMT=0
4400
CLS:LOCATE27,12:PRINT"Balancing the
Checkbook"
```

```
4510 LOCATE8,10:PRINTC+FE"Checks and Debits Totaling";:LOCATE60,10:
PRINTUSING" $$##,###.##";-B2+-BC
4520 LOCATE8,12:PRINT"Your Bank Statement Balance should Read ";:
LOCATE60,12:PRINTUSING"$$##,###.##";B3
4530 LOCATE8,14:PRINT"Your Checkbook Balance is";:LOCATE60,14:PRINTUSING"
$$##,###.##";B3+B4
4540 LOCATE0,18:PRINTSTRING$(80,"$");
:LOCATE0,23:ATTR2,7:PRINTLC$;
4550 LOCATE22,23:PRINT"Press any Key to Return To Menu";
4560 EXEC44539:GOTO9010
5000 '** UPDATE ROUTINE **
5010 ENT=0:DE=0:AMT=0:N=0
5020 CLS:GOSUB 7000
5030 GOSUB7500
5040 LOCATE2,2:PRINT" Date Check No. Amount Paid To Exp. Cat. Clear No."
5050 LOCATE0,0:ATTR2,1:PRINTLC$;
5060 LOCATE50,0:ATTR2,1:PRINT MC$;:LOCATE3,0:PRINTWB$" Checks";:LOCATE44,0:INPUT"CHANGE or DELETE which Entry ";EN$:
ATTR2,2:LOCATE0,1:ENT=VAL(EN$)
5070 IF ENT=>LOF(1)+1 OR ENT=0 THEN CLS:GOTO5040
5080 GET#1,ENT
5090 LOCATE0,4:PRINT" "DA$: LOCATE 14,4:PRINTCH$:LOCATE21,4
5100 IFEC$="DD"OREC$="II"THEN PRINTUSING"$$##,###.##";CVN(AMT$):
GOTO5120
5110 PRINTUSING"$$##,###.##";-CVN(AMT$)
5120 LOCATE36,4:PRINTPA$:LOCATE 60,4:PRINTEC$:LOCATE70,4: PRINT CL$:
LOCATE74,4:PRINTENT;
5130 LOCATE26,6:ATTR2,0:PRINT "CHANGE or DELETION <C or D>";
5140 K$=INKEY$:IFK$=""THEN5140
5150 IF K$="C" THEN 5160 ELSE IF K$="D"THEN 5500 ELSE5140
5160 LOCATE0,4:ATTR2,1:PRINT"1";:
LOCATE11,4:PRINT"2";:L OCATE20,4:
PRINT"3";:LOCATE33,4:PRINT"4";:LOCATE 57,4:PRINT"5";:LOCATE67,4:PRINT "6";:LOCATE0,5:ATTR2,2
5170 LOCATE0,23:ATTR2,0:PRINTLC$;:
LOCATE28,23:PRINT"Press BREAK for MENU";
5180 D$=DA$:C$=C H$:AMT=CVN(A MT$):
AMT=CVN(AMT$):P$=PA$:E$=EC$:V$=CL$
5190 LOCATE26,6:ATTR2,6:PRINT" Select <1 to 6> ";
5200 K$=INKEY$:IF K$=""THEN5200
5210 IFVAL(K$)<1 ORVAL(K$)>6THEN5190
5220 LOCATE26,6:ATTR3,2:PRINT "Correction appears in WHITE";:
LOCATE3,8:ATTR2,4
5230 ON VAL (K$) GOTO5250,5270, 5290,5330,5350,5370
5240 GOTO5200
5250 LINEINPUT"New Date -> ";D$
5260 LOCATE2,4:ATTR3,2:PRINTD$;:
GOTO5390
5270 LINEINPUT"New Check No. -> ";C$
5280 LOCATE14,4:ATTR3,2:PRINTC$;:
GOTO5390
5290 LINEINPUT"New Amount -> $";
A$:AMT=VAL(A$)
5300 IFEC$="DD" OR EC$="II"THEN AMT=AMT ELSE AMT=-AMT
5310 LOCATE21,4:ATTR3,2:IFEC$="DD" OREC$="II"THENPRINTUSING" $$##,
###.##" ;AMT;:GOTO5390
5320 PRINTUSING"$$##,###.##";-AMT;:
GOTO5390
5330 LINEINPUT"New Paid To -> ";P$
5340 LOCATE36,4:ATTR3,2:PRINTP$;:
GOTO5390
5350 LINEINPUT"New Expense Cat. -> ";E$
5360 LOCATE60,4:ATTR3,2:PRINTE$;:
GOTO 5390
5370 LINEINPUT"New Clear Code -> ";V$
5380 LOCATE70,4:ATTR3,2:PRINTV$;
5390 ATTR2,2:LOCATE3,8:PRINTMC$:
LOCATE0,1
5400 LSET DA$=D$:LSET CH$=C$:LSET AMT$=MKN$(AMT):LSET PA$=P$:LSET
EC$=E$:LSET CL$=V$
5410 PUT#1,ENT:LOCATE0,23:ATTR2,7:
PRINTLC$;
5420 LOCATE25,23:PRINT"Any More Corrections <Y or N> ?";
5430 K$=INKEY$:IF K$=""THEN5430
5440 IF K$="Y"THEN LOCATE26,6:
ATTR2,2: PRINT MC$;: GOTO5450 ELSEIF K$="N"THEN9010 ELSE5430
5450 LOCATE0,23:ATTR2,2:PRINTLC$;:
GOTO5060
5500 '***** DELETE ROUTINE *****
5510 LOCATE 26,6:ATTR2,7:PRINT" ARE YOU SURE <Y or N> ? ";
5520 K$=INKEY$:IFK$=""THEN5520
5530 IFK$="Y"THEN5540 ELSEIF K$="N"THEN9010 ELSE5520
5540 OPEN "D",#2,"NEWCHKS.DAT",45
5550 FIELD#2,9AS ND$,5AS NC$,5AS NA$,20AS NP$,3AS NB$,2AS NE$,1AS NV$
5560 DE=ENT:IF ENT=1 THEN5630
5570 FOR ENT=1 TO DE-1
5580 GET#1,ENT:NAMT=CVN(AMT$)
5590 LSET ND$=DA$:LSET NC$=CH$:
LSET NA$=MKN$(NAMT):LSET NP$=PA$:
LSET NB$=BN$:LSET NE$=EC$:LSET NV$=CL$
5600 N=N+1:PUT#2,N
5610 IF N>DE OR N=DE THEN5670
5620 NEXT ENT
5630 FOR ENT=DE+1 TO LOF(1)
5640 IF ENT>LOF(1) THEN5680
5650 GET#1,ENT
5660 GOTO5580
5670 NEXT ENT
5680 CLOSE
5690 LOCATE26,6:ATTR2,1:PRINT" RECORD "DE;" IS DELETED ";:ATTR2,2
5700 KILL FI$+"CHKS.DAT"
5710 RENAME"NEWCHKS.DAT" TO FI$+"CHKS.DAT"
5720 GOTO9010
6000 '** PRINT OUT ROUTINE **
6010 IF (PEEK(65314)AND1)=1 THEN6650 ESLE 6020
6020 PRINT#-2,STRING$(80,"$")
6030 IF A$="A" THEN 6040 ELSE 6050
6040 PRINT#-2:PRINT#-2,TAB(26);WB$;
"Checks":PRINT#-2:GOTO6060
6050 PRINT#-2:PRINT#-2,TAB(20);WB$;
"Checks for Month ";WM$:PRINT#-2
6060 PRINT#-2,TAB(2)"DATE";TAB(9)
"CHECKNO.";TAB(21)"AMOUNT";TAB(39)
"PAID TO";TAB(56)"CAT";TAB(63)
"BALANCE"; TAB(72)"CLR";TAB(76)"NO."
6070 PRINT#-2,STRING$(80,"-"):RETURN
6080 PRINT#-2,DA$;TAB(11)CH$;TAB(17);
6090 IFEC$="DD"OR EC$="II"THEN PRINT#-2,USING "$$##,###.##";CVN
(AMT$);:GOTO6110
6100 PRINT#-2,USING"$$##,###.##";-CVN(AMT$);
6110 PRINT#-2,TAB(33)PA$;TAB(56)EC$;
TAB(60);:PRINT#-2,USING"$$##,###.##";
BAL;:PRINT#-2,TAB(73)CL$;TAB(75)E:
RETURN
6120 PRINT#-2,STRING$(80,"-"):PRINT#-2
6130 PRINT#-2,TAB(20)"Checkbook Balance is";:PRINT#-2,USING "$$##,
###.##";BAL
6140 PRINT#-2:PRINT#-2,STRING$
(80,"$"):RETURN
6500 '** EXPENSE PRINTOUT **
6510 IF(PEEK(65314)AND1)=1 THEN6650 ESLE6520
6520 PRINT#-2,STRING$(80,"$")
6530 PRINT#-2:PRINT#-2,TAB(17)WB$;
"Checks for Expense Category ";E$: PRINT #-2
6540 PRINT#-2,TAB(5)"DATE";TAB(16)
"CHECK NO.";TAB(32)"AMOUNT";TAB(53)
"PAID TO";TAB(76)"N0."
6550 PRINT#-2,STRING$(80,"-"):RETURN
6560 PRINT#-2,TAB(3)DA$;TAB(19)CH$;TAB(29);
6570 IFEC$="DD"OR EC$="II"THEN PRINT#-2, USING "$$##,###.##";CVN
(AMT$);: GOTO6590
6580 PRINT#-2,USING"$$##,###.##";-CVN(AMT$);
6590 PRINT#-2,TAB(47)PA$;TAB(75)S:
RETURN
6600 PRINT#-2,STRING$(80,"-"):PRINT #-2
6610 IF PE$="A" THEN 6620 ELSE 6630
6620 PRINT#-2,TAB(26)"Category "E$" Total is";:PRINT#-2,USING"$$##,###.##";
BAL:GOTO 6640
6630 PRINT#-2,TAB(23)"Total for Month "PE$;" is";:PRINT#-2,USING "$$##,
###.##";BAL
6640 PRINT#-2:PRINT#-2,STRING$
(80,"$"):RETURN
6650 CLS:LOCATE 30,12:PRINT"Printer Not Ready"
6660 FORT=1TO1000:NEXTT:CLS:
GOTO210
7000 '** EXPENSE CATEGORY CODES **
7010 LOCATE0,18:ATTR2,5:PRINT"
Advertising = AD Office Supp. = OF Resale Supp = RS Printing = PR"
7020 PRINT" Shipping = SH Travel = TR Taxes = TX Miscellaneous = MI"
7030 PRINT" Computer = CM Deposit = DD Royalty/Writer = RW"
7040 PRINT" Credit Card = CC Non-Business = NB"
7050 REM PRINT" Phone = PH Prop. Tax = PT Rent = RE Spending = SP"
7060 REM PRINT" Dr./Dentist = DR Business = BS"
```

continued on page 9

Roadmap to the Internet, Part 3

Patrick D. Crispen

E-mail and Listservs

"I have received no more than
one or two letters in my life that
were worth the postage"
— Henry David Thoreau

Almost all e-mail programs have similar, universal functions. The problem is that all use completely different commands to access these functions (example: to reply to the author of a current message using the elm or pine e-mail programs, you type the letter "r", to do the same function in the VM Mailbook program you have to hit the F5 key).

I'm not going to be able to discuss all of these functions, but what sort of functions do most e-mail programs have in common? Well, most mail programs have a function that will allow you to access and read your incoming mail, another to save incoming mail in a file, one to print incoming mail, one to send new messages, one to reply to a message, another to include a file in a mail message, and one to import/export special objects into your mail messages. Depending on your e-mail software, these functions are either easy or difficult ... but nearly always possible.

With all of the different e-mail programs out there, and all of the different commands required to run each program, how are you ever going to find out what commands are right for YOUR e-mail program? Easy! Ask your local e-mail service provider! This may shock you, but almost every mail provider provides some sort of instruction sheet or file that will teach you how to use the e-mail program that your provider is running. All you have to do is ask!

I want to take a moment to show you how to actually read an Internet address. I have to admit that when I first started learning how to use e-mail, I was intimidated by the length of all of the Internet addresses. However, once I learned to read the addresses BACKWARDS — from right to left — Internet addresses ceased to be a thing of mystery.

Sample Internet Address (mine):
PCRISPE1@UA1VM.UA.EDU

Every Internet address has three parts — a user name, an "at" sign (@), and the address of the user's mail server. In this example, my user name is PCRISPE1 (and stop laughing — there is nothing funny about "p-crispy-one"), and my mail server's address is UA1VM.UA.EDU. The mail server address (the UA1VM.UA.EDU part of the above example) is actually called the "domain" name, and it is based on something called an IP (or Internet Protocol) address.

Each server connected to the Internet has a numerical IP address. The IP address is four sets of numbers connected with periods (for example, the IP address for the mail server that I am using at the University of Alabama is 130.160.4.100).

Fortunately, the powers that be realized that people remember NAMES better than numbers, and they created the domain name system. The domain name system associates the numerical IP address with an easier to remember "name" (for example, thanks to the domain name system, the IP address 130.160.4.100 becomes a much easier to remember UA1VM.UA.EDU).

You may run into IP addresses from time to time when you are FTPing or telnetting (we'll talk about both of these tools in a few weeks). Just remember that an IP address (the four sets of numbers connected with periods) is simply another way to write a domain name, and you will do fine. Both IP addresses and domain names should work equally well.

Anyway, back to the "p-crispy-one" example. Remember that my domain name is UA1VM.UA.EDU? Well, as I said earlier, the best way to read an Internet address — and, for that matter, a domain name — is from right to left. Domain names are broken down as follows:

EDU Educational sites in the U.S.
COM Commercial sites in the U.S.
GOV U.S. Government sites
NET Network administrative organizations
MIL U.S. Military sites
ORG U.S. Organizations that don't fit into other categories
SU Soviet Union (yes, there is still a Soviet Union ... at least on the Internet)
FR France
CA Canada
(other counties have their own country code)

Since my domain name has an EDU at the end of it, we now know that UA1VM.UA.EDU is the domain name for some educational site in the United States. But where? The rest of the UA1VM.UA.EDU domain name lists the "subdomains" that tell you where my mail server is actually located. UA is the University of Alabama, and UA1VM is the name of my mail server's machine.

So, PCRISPE1@UA1VM.UA.EDU is the Internet address for someone named "p-crispy-one" (stop laughing!!) at some U.S. educational site. Further investigation shows that the site is at the University of Alabama, and that the machine "p-crispy-one" is using is called UA1VM.

Another Sample Internet Address: w.v.braun@hq.msfc.nasa.gov

Okay, reading this right to left, we see a GOV. That means it's a U.S. Government address. I think we all know what NASA is — the National Aeronautics and Space Administration. Unless you are a big NASA fan, however, you probably don't have the slightest clue what MSFC stands for (it is the George C. Marshall Space Flight Center in Huntsville, Alabama). HQ is pretty self-explanatory — Headquarters.

So we know that w.v.braun@hq.msfc.nasa.gov is the address of some person named w.v.braun whose mail server is at the Headquarters of the Marshall Space Flight Center, and that the Marshall Space Flight Center is part of NASA, which itself is part of the U.S. government.

What can you tell from this Internet address: ike@saceur.pentagon.army.mil

A lot, especially if you are a history buff, and if you know that "saceur" is the military abbreviation for Supreme Allied Commander-Europe (this is a phony address, by the way!).

The best rule of thumb I can give you about Internet addresses is this: if the address is not of the form described above and does not end with one of the standard top-level domain abbreviations or country codes, the address is NOT an Internet address. You may still be able to send mail to non-Internet addresses through a gateway, though.

HOMEWORK:

Today's homework is completely optional. Remember though, please do not send your homework assignments to me :)

Also, please remember that replying to this (or any other) letter with your GET commands will *NOT* work. You *MUST* send your GET commands in the body of a *NEW* e-mail letter to LISTSERV@UA1VM.UA.EDU

1) I want you to find the following two commands for your mailer:

- the command that allows you to delete an e-mail letter without having to read the letter
- the command that allows you to delete an e-mail letter after you have read the letter

You will soon discover that these two commands are the most important, and most used, e-mail commands you will ever use.

The DELPHI on-line mail system is VMS based (see #2 below). For instructions about your off-line reader refer to the appropriate manuals. Even if you are using an Off-Line Reader you should learn to use the DELPHI on-line mail system in case you need to retrieve mail you have previously downloaded and lost (e.g. if you accidentally deleted them). See "Delphi Mail Commands" later in this article.

2) If you have "Level Two" or "Level Three" connectivity and are on a UNIX, VAX/VMS, or VM system, there are three files I want you to GET from the University of Alabama's LISTSERV file server (see a previous issue's LISTSERV FILE SERVER COMMANDS for a review of the GET command). The files are from Richard Smith's "Navigating the Internet" workshop, and Richard was kind enough to give me permission to use them in this workshop.

The first file covers the basic e-mail commands for the UNIX, VAX/VMS, or VM systems. The second file covers the commands to send e-mail, and the third file covers the reply function. Remember when using the GET command that your commands must be sent to the LISTSERV address, not to the list or to me.

UNIX USERS:		VAX/VMS USERS:	
filename	filetype	filename	filetype
UNIX	1	VMS	1
UNIX	2	VMS	2
UNIX	3	VMS	3

VM USERS:

filename	filetype
VM	1
VM	2
VM	3

You will have to use three GET commands (one for each file), but you can put all three GET commands in one letter. For example, if I wanted to get all three of the VM files, the body of my letter would look like this:

GET VM 1 F=MAIL
GET VM 2 F=MAIL
GET VM 3 F=MAIL

PLEASE REMEMBER TO SEND YOUR GET COMMANDS — OR *ANY* OTHER LISTSERV COMMANDS — IN THE *BODY* OF AN E-MAIL LETTER SENT TO:

LISTSERV@UA1VM.UA.EDU

3) If you are not on a UNIX VAX/VMS, or VM system — or if you are not sure what sort of system you are on — contact your local Internet provider and ask for some information on how to use your mail program.

In particular, you should ask for information on how to:

- access your e-mail program
- open and read e-mail sent to you
- save an e-mail letter to a file
- print an e-mail letter
- send a new e-mail letter to someone
- reply to an e-mail letter sent to you
- include text in a reply (and how to edit this text)

You probably know how to do most of these things, but it never hurts to review it from time to time.

4) If you would like to get a list of all of the Internet Country Codes, use the GET command to get the file COUNTRY CODES from the University of Alabama's LISTSERV file server.

Delphi Mail Commands

If you are using an off-line reader such as Info-Express you may never need to use these commands. However, they are helpful to know in those cases you receive a message while on-line or if you accidentally delete a message from your OLR. Mail messages on DELPHI remain available for several weeks unless you actively delete them.

The following material appears in the DELPHI mail system help files. To use the help files simply type HELP and press ENTER while in the mail system. To few this information select the GETTING_STARTED topic.

To become familiar with the Mail Utility, use the MAIL commands discussed in this section. These commands enable you to move around within the Mail Utility. For more detailed information about the Mail Utility, see the VAX/VMS Mail Utility Reference Manual.

These are the MAIL commands discussed in this section:

SEND DIRECTORY EXTRACT
READ[/NEW] DELETE PRINT
FORWARD MOVE HELP
REPLY SELECT EXIT

The first command to try is the SEND command. Try sending a message to yourself. Enter the SEND command and press RETURN. Enter your own user name at the prompt and press RETURN. Enter a subject at the prompt and press RETURN again. The following example shows how to use the SEND command:

MAIL> SEND
To: PIERCE
Subj: SAILING
Enter your message below. Press CTRL/Z when complete, or CTRL/C to quit

When you finish entering the text of your message, press CTRL/Z. Because you are sending the message to yourself, MAIL signals that you have just received a new message by displaying the following message:

New mail on node FLAXEN from PIERCE
MAIL>

Now, you are ready to use the READ command. To read the message you just sent to yourself, enter the READ command with the /NEW qualifier and press RETURN as follows:

MAIL> READ/NEW

You must specify the /NEW qualifier with the READ command when you want to read new mail that arrives while you are in the Mail Utility. When you are not in the Mail Utility and you receive new mail, invoke MAIL to read the new message, you can enter the READ command without the / NEW qualifier. Or, if you wish to read mail that you have already read, you can enter the READ command.

You can forward a copy of a mail message to another user by entering the FORWARD command. MAIL prompts you for the name of the user to receive the message. Try forwarding a copy of the message you just received back to yourself. Enter your own user name and press

RETURN. Supply a subject when prompted and press RETURN. MAIL signals that you have just received a new message. Enter the READ/NEW command to read the forwarded message.

When you receive a message and want to respond to it, enter the REPLY command and press RETURN. MAIL displays the header information as follows:

MAIL> REPLY
To: FLAXEN::PIERCE
Subj: RE: Using the REPLY command
Enter your message below. Press CTRL/Z when complete, or CTRL/C to quit:

When you finish typing your response, press CTRL/Z. Again, MAIL signals that you have just received a new message. To read the message, enter the READ/NEW command.

When you want to see a list of all the mail messages you have collected, enter the DIRECTORY command and press RETURN. MAIL displays a list like the following:

#	From	Date	Subject
1	FORBES	1-JUN-1988	How to Write a Memo
2	STELLA::BERT	2-JUN-1988	Using the Printer
3	FROST::BASTIEN	4-JUN-1988	Chicken Kiev

When you want to remove a message, use the DELETE command. You can either enter the DELETE command while you are reading the message or you can enter the DELETE command followed by the number of the message you want to remove. To remove the second message in the list, enter the following command line:

```
MAIL> DELETE 2
```

If you enter the DIRECTORY command after you have deleted a message (or messages), you see the messages marked for deletion, as follows:

```
# From          Date          Subject
1 FORBES        1-JUN-1988    How to Write a Memo
2 (Deleted)
3 FROST::BASTIEN 4-JUN-1988   Chicken Kiev
```

When you exit from MAIL, the messages marked for deletion disappear.

The Mail Utility allows you to organize your messages by moving them into folders. To move a message to a folder, enter the MOVE command (while you are reading the message) and press RETURN. MAIL prompts you for a folder name. Type any name, for example, REVIEWS or JOKES or STATUS_REPORTS. MAIL also prompts you for a file name. You can specify the default mail file by pressing RETURN. A sample session:

```
MAIL> 2
MAIL> MOVE
_Folder: WINNERS
_File: <RET>
Folder WINNERS does not exist.
Do you want to create it (Y/N, default is N)? Y
%MAIL-I-NEWFOLDER, folder WINNERS created
```

In this example, the folder name is WINNERS and the default mail file is specified. If the folder you name does not exist, MAIL asks if you want to create it.

Once you have created folders, you may want to move between them. To move from one folder to another, use the SELECT command. If you want to move to the WINNERS folder, enter the SELECT command as follows:

```
MAIL> SELECT WINNERS
 %MAIL-I-SELECTED, 1 message selected
```

In this example, MAIL displays a message indicating the number of messages in the folder.

To move to a folder named JOKES, enter the following command line:

```
MAIL> SELECT JOKES
%MAIL-I-SELECTED, 32 messages selected
```

You can enter the DIRECTORY command to see a list of the messages in the folder you just selected.

When you want to move a mail message from your mail file to a sequential file that you can access from the DCL command level, use the EXTRACT command. Enter the EXTRACT command (while you are reading the message) and press RETURN. MAIL prompts you for the name of a file. Then, when you exit from MAIL, the file is listed in your directory. The following example shows how to use the EXTRACT command to move a mail message to a file named GAMES.DAT.

```
MAIL> EXTRACT
_File: GAMES.DAT

%MAIL-I-CREATED, DISK: [BERGMAN] GAMES.DAT;1 created

MAIL>
```

To print a hard copy of a mail message, enter the PRINT command while you are reading the message and press RETURN. (When you exit from MAIL, the message enters the print queue.) The following example shows how to make a hard copy of message #4 by using the PRINT command:

Please be aware that using the PRINT command will NOT print on your printer, it will print on DELPHI's printers in Cambridge. Therefore, you should normally never use this command.

```
MAIL> 4

#4  4-AUG-1988 09:39:20  MAIL
From: SPARTA::FELLINI
To:   MARSTON
Subj: Rydell's Reasons
In reference to the meeting of July 26, I would like to explain
Rydell's opinion more fully...

MAIL> PRINT
```

You have already started using the MAIL HELP command. The next HELP topic to look at is "Folders". "Folders" discusses the organization of the Mail Utility in detail. For more information about each MAIL command, you can keep using the HELP facility provided in MAIL or you can see the VAX/VMS Mail Utility Reference Manual.

When you are ready to leave MAIL, enter the EXIT command and press RETURN. Any messages marked for deletion disappear. Any messages marked for printing enter the print queue and the following message is displayed:

```
MAIL> EXIT
 Job MAIL (queue ATLAS_PRINT, entry 43) started on QUEUE$LPA0
```

Patrick D. Crispen is a professor at the University of Alabama. This series was originally taught over the Internet itself. It has been modified to fit a printed format with as few changes as possible

< 268'm >

No Hassle Check Register

continued from page 6

```
7070 LOCATE0,23:ATTR2,2
7080 RETURN
7500 '** OPEN FILE **
7510 OPEN"D",#1,FI$+"CHKS/DAT",45
7520 FIELD#1,9AS DA$,5AS CH$,5AS AMT$,20AS PA$,3AS BN$,2AS EC$,1AS CL$
7530 RETURN
8000 '** NOT IN FILE **
8010 LOCATE25,12:PRINT"Requested Data not in File";
8020 FOR T=1TO900:NEXTT
8030 CLS:GOTO210
8500 '** QUIT **
8510 CLS:LOCATE25,13:PRINT"ARE YOU SURE? (Y/N)":EXEC44539
8520 A$=INKEY$
8530 IF A$="N"THEN 8540 ELSE 8550
8540 CLS:GOTO 210
8550 IF A$="Y"THEN 8560 ELSE 8510
8560 CLS:POKE113,0:EXEC40999
9000 '** ON BREAK GOTO **
9010 CLOSE:UNLOAD:ATTR2,2:CLS:IF FI$=""GOTO160 ELSE210
```

< 268'm >

Operating System Nine

Rick Ulland

Multi-Vue

MultiVue is, well, different. Compare with other current GUI's- System7 is, well, a System, and while traditional Windows starts out as a menu, it also attempts a major fix on DOS, while it's running under it (no wonder they always had problems with <4.0- note the jury is still out on Windows92, as it's just been released). In any case, the gui is intricately tied to the system, rigidly structured, and well documented. Under OS9, we already have a good multitasking opsys with multiple windows- in theory, a little window dressing is all it takes to make a decent gui. It is, _Provided_ the OS9 system is organized exactly as the authors expected.

Standard disclaimer: Any MVue user will tell you, the gshell 1.24a (Kent Meyer's patch) is much nicer (and more reliable) than stock. See the sidebar for installation tips.

MultiVue was written to manage a typical OS9 environment, as envisioned by MicroWare/Tandie. In this vision, everybody had a hard drive and needs MultiVue to manage the hundreds of files this creates. Not entirely beside the point, a hard drive is fast enough to shuffle the extra bytes- a directory structure +AIF +ICON makes one big database! Floppy support almost looks tacked on - limited enough it's a pain to maintain the even strain. much basic information has to repeated across most floppies and the load time is abominable. It's usable, especially if you need the point and click- but let's start with the easy example of a hard drive system.

The idea is to create a primary level of systems directories, including /dd/cmds (every OS9/6809 program ever recorded), /dd/sys (every environment and termset before, but not including, vt100), and language/storage dirs like /dd/docs, /dd/lib, /dd/defs, ad nauseum... you'll find there is pretty much a consensus on what these dirs are, and in fact if you simply dsave all your original floppies to a hard drive (chd /d0; dsave /d0 /h0 ! shell) this structure will sort of 80% form by itself. This leaves the root directory full of 'folders' you didn't name and can't hide or move. Unfortunate, as MultiVue always displays the root directory first. But remember, these dirs are for the machine. They should be organized to suit it- as example /dd/cmds is _supposed_ to have 750 entries. You'd think it would make sense to split things up, especially if you sort and your favorite program is zzzzyzwz. Don't bother. It takes more time to maintain a non-standard organization than it ever saves in load- and you will never personally have to go to cmds if you play the cards right.

Prenote: The major difference between floppies and hard drives is the presence of a /USR directory branch. Floppies are just to small to bother with the added complications- if there is room for any data, you put it in root+1 at best. Although some companies, like CoNect, do add /USR/ETC to some distribution floppies to make hard drive installs easier, you don't want to run that way on flops (understand me later but read me now when I say.....)

This distinct /USR subtree makes lots of sense with multiple users, where you have / usr/sid/downloads and /usr/joe/mail... and works out pretty good for MultiVue as well. In /dd/usr, place 4 or 5 directories that describe your interests, or otherwise split the combined total of 'your stuff' up in some logical way. Usually, this is _not_ by program, instead a dozen or so programs used to preform some task will be assigned to a directory- /dd/usr/desk, /dd/usr/gfx, etc. This 5 folder usr screen will pop up rapidly, and each folder will also be smaller and quicker loading. If some primary interest gets large, split that branch again. In fact, the taller the tree gets, the better MultiVue runs.

A helpful trick is to hide data where MultiVue doesn't have to display it. There are already some directories you don't look at, like cmds. Some authors have 'hidden' their programs data- for instance, MVCanvas loads it's own data, from /dd/pics. There's no particular reason to run MVCanvas from there, so it's icon can be elsewhere- say the USR/GFX directory. There won't be any MVCanvas data files here, just the icon to start it- a fast loading directory screen, so far. Most programs don't do this for you, and (scrape) the ones that do should pick a subdir- I vote for an APPS branch after CMDS and SYS.

Anyway, we are left with some programs that insist on putting data in the current data dir, so let's extend the hidden/visable concept with a new directory type, specifically to look at- the 'lookat' is an artificial directory level specifically designed to make a pretty MultiVue screen. The data lies in the unseen levels above it, and what MVue don't see can't slow it down.

For programs that load their own data, the quick fix is to stick the data in a subdir of lookit, leaving the icon where it was. After clicking the program to life from lookit, change the programs data directory using it's own menus, or simply expand the typed in filename into an extended path('subdir/file.ext' without a beginning slash).Note that it's as easy to back up one level. Say you are now in a subdir, and need your sigfile.doc from lookit. Simply load ../sigfile.doc (dot is where you are now, dot dot is the dot before dot.... previous menu, whatever- Huh?- try dir . and dir ..)

Another solution is to write a shellscript that moves to the new dir before running the program. (Details to follow- afraid I've been writing to much html to form a complete paragraph anymore) When you quit the program, you drop back to the original lookit. If you don't mind the added step, you can design directory screens that have nothing to do with the above structure and jump all over the disk once clicked. Put them all in root (AIF in caps, for the alpha sort)- Granny Smith? This can make for some pretty MultiVue screens, but it's all too easy to lose data since it's not really connected to the programs that generated it..

An example: Originally, all telecom was done from /dd/usr/tcom, which was set up to do simple email, download/extract files, and so on. When the 'web' struck, it created all sorts of data that didn't fit there, so /dd/usr/tcom/web was created to handle all the things I didn't previously have to handle. Read as a menu, once you select user, then select tcom, there is a web folder available....

When Infoexpress came along, it's data structure was huge and multivue unfriendly. A mailbox icon was added to the root dir as a hotkey, which called a one line basic09 procedure (SHELL"cd /dd/usr/tcom/ix/ix;ex shell ix #1 <>>>/1"). Study the cd (chd) part. First, this is not a random location, instead it's on the obvious /dd/usr/tcom branch. Doesn't have to be, but if the script gets lost, that's where I'd look! Note there are two ix diectories- if the user climbs up though tcom to the first ix, there is a lookit with text editors, uucode and the like, designed to create external files for later posting from Infoexpress. There is a second IX directory visable, but the fact it's not a new idea suggests there is no reason for a MVue user to go there. The lookit also contains a copy of the 'hotkey' mailbox AIF (from the root dir) which will still work, since the script it runs cd's an absolute pathlist this AIF could be copied anywhere- and clicking it moves one to the 'invisible' ix/ix dir, running Infoexpress and it's 396 datafiles. The 'file send' files are simply named ../filename when sent/received, which looks 'back' to it's parent lookit (first IX) directory. [A little more example than I really intended.....<wri>]

This sort of organization may seem a little alien, especially if familiar with MSDOS and it's organize-by-program mentality. It may help to think of the directory structure as a menu- consider the following (long) pathlist- /dd/usr/conect/product/fast232/diskimage/modules/t2.dd I can find this descriptor at will, since each step proceeds logically from the previous, which is how I can remember (actually recreate) such a long pathlist on the fly. A flatter directory structure would make a shorter line to type, but you'd have to remember locations by route, and the new user would never find the file without an index. Instead, we have a logical menu path- which also makes sense when viewed as a series of MultiVue screens if user knows to try USR first. The 'menu system' isn't part of MultiVue at all, but standard OS9 disk organization- a gopher menu tree is similar.

From the command line, this sort of disk structure will help find a data file, but there are no particular prompts as to which programs might be helpful after you find them. All programs are available all the time. With the AIF system, you can prompt your users with a half dozen suggestions. These short text files do

many things- first, an icon is selected to represent the program. Since most programs have their own distinct file extention, (ie: dynacalc makes *.cal files) they also link programs with the data they create- AIF.cal links all *.cal files to the program named within. If an AIF of a data file with known extention is clicked, we next set up a screen type/color, assign the program some RAM and start it, possibly with a filename to load.

Loading all of this information at one time would cause several problems. There would be too much data to get at once, and no way to handle two programs which used the same extention. Instead, AIF's are loaded as the user moves up the dir tree- there is no reason you can't have seven different types of *.cal files, provided there aren't two different types in the same directory branch. The buffer is flushed when user returns to the drive bar to start another run up the tree. Note this buffer is *not* flushed when the path is reversed using the 'backup box', which means you can drag AIFs back down the tree. This can cause some strange problems- if an icon sometimes runs the wrong program, look for an earlier AIF with the same name.

Consider what this means to you, the sysadmin. It's faster to not load something than load it, so it's nice to push most of the AIFs to the top, with the data files. But this can leave lots of duplicated AIFs, for example a word processor is used almost everyplace. Putting a few much used AIFs directly in USR (before the lookats) means they will be available in all the lookats. You'll still have to duplicate some AIFs- there has to be an AIF or at least one preexisting data file in a lookit, or there's nothing to click.

A example- lets assume the user programs are in /dd/usr/.... and you want any .doc file in this branch to start a text editor. But, the manual pages in /dd/docs should instead run a text viewer, even though the pages are also .doc files. The solution would be to put the text editor AIF.doc in /dd/usr (or later), and put the viewer AIF.doc in /dd/docs. Depending on the path the user climbs from /dd, a different AIF.doc will be loaded. If you need to edit a doc file, just travel to /usr (picking up the text editor AIF) then close usr by clicking the tiny box at the start of the pathlist (keeping the AIF) and then go to /docs.

AIF's don't have to start a specific program. Name 'shell' or 'basic09' as the program, and specify the RAM and screen type you want. Using several AIFs, each with shell or B09 as program and a different icon to represent the screen type created, makes an easy way to 'pop' the most used window types. Note that gfx screens have a special feature- if the screen size given in the AIF is smaller than a full screen of the type given, this is taken to be the minimum size, and a sizing box is displayed. The user can expand this up to a full screen, or leave the minimum size you allowed. I used to use 32x16 (minimum) on an 80x24....

One last organizational trick- some programs don't really need to be in a data directory (like the mailbox AIF, above). You can put a few dozen of these in the root dir and make a Mac looking 'all important programs here' window, which even pushes the machine directories offscreen. Works great, as long as Multivue is in the root dir. But what if you want to run a program from a lookit, without losing the bomb screen?

Use an AIF to call MultiVue (actually gshell) itself. Drive this second MultiVue anyplace you want, and leave the original screen in root. It's interesting this works, because attempting to start two MultiVues from the same shellscript (as in a startup) gives the second one a dead menu bar. Doesn't happen from an AIF (very often), and more- if an ill behaved app like shellmate clobbers your original gshells menu bar, starting a second MVue fixes the original gshell, as well as working itself- but this clobbers ShellMate(etc)'s menu bar... (think gsort). Also, you need an AIF to start control on a 16 color screen!

Despite their usefulness, AIF's are pretty simple minded. They'll do everything exactly once. There are ways to make them smarter- for instance you could call a shell script, and do quite a few things before eventually running the main program. As example, MultiVue doesn't 'do' the vdg screens used by Tandy games. So the game patch AIFs usually call a shellscript to pop the VDG window, change the dir, run the game concurrently, and exit (killing the useless screen the AIF created to run the script shell in). Not all AIF driven scripts are that complex, most merely chain or pipe a few programs together.

You'll need Shell+ for anything fancy - refer to the shell+ docs and this column (v2 #6 = Mar95) for details. Also, check out the game patches (available everywhere), they _are_ clever.

Shellscripts commonly look a little ugly under MVue, due to the way an AIF launches programs. You have to specify 'shell' then 'scriptname' as parameter, and your icons have 'shell' written under them instead of the clever name you thought up for the script. One workaround is to take advantage of runb. If you name a 'packed' B09 icode module on the command line, OS9 has the smarts to figure this is really a parameter and runb is the actual program. And MultiVue is stupid enough to accept the 'program' name on blind faith. Now you can name your terrific script by name, even if your 'basic' program is 50 lines of SHELL"something". If still using the stock shell (shame!) these programs also have the advantage of fitting in the exec dir (Shell+ shellscripts can also be in exec dir, but stock has to clutter them in current data) Note that execution drops back to Basic09 between each line. If you want to chd, chx, and run a program, you'll have to do it on one line, so it all goes to the same shell.

SHELL "chd /h3/pub/this; chx /h6/cmds; barftp&"

Random Thoughts

When an AIF calls a program, it's RAM allocation overrides shell's default- even when you leave the RAM line blank, MVue's default covers shell's default. Which means, if you've patched shell+ to change the default RAM allocation, it goes away when an AIF calls the shots- which can be both good and bad.If a program will work from the command line and not an AIF, this is one thing to try. If you really need to squeeze RAM, you can assign as little as a page from the AIF, which is what some of the patches do to shell's default. Further note: If you start shell+ from an AIF, guess how much RAM it's got?

MultiVue (patched) can sort directories alphabetically, caps first. If you name your AIF's in caps ('AIF.stuff' even though 'aif.stuff' is allowed) and name datafiles in all lower case, the screen will sort to iconed programs, folders, then datafile icons.

Best Buy- a hard drive. If you're this interested, you deserve one. 10 times faster is 10 times funner, and we ain't touched bigger yet.

Suggested Writing- MultiVue needs a single preloaded 'Desktop' screen to replace the lame Tandy menu. This screen would preload the dozen AIF's described above as 'Root hotkeys' from some default dir and act as an omnipotent program launcher- starting any program in it using the current data dir.

Floppies-

So you're stuck with a few floppies. More difficult, but not impossible. Floppies are _very_ slow, so you'll still need to divide and conquer. You might want to keep a usr directory, with a few subdirs, each no bigger than 6 icons and a dozen data files. This may appear picky small, but chasing up a menu tree is faster than floppyloading 50 icons (or nonicons) from a big root..

Installing MultiVue.

MultiVue has a presumably handy self-installer, but you probably know more about OS9 boot by now than the installer does. With EZGen or Kwickgen, or a valid OS9Gen bootlist, do this:

First, the GrfInt window driver should be replaced with the menuing WindInt version. If you are planning to run Tandy games, you'll need VDGInt also- which can make a tight fit, especially if VRN is used to work in the big games. Bug or feature? The MultiVue user boot for a 'full on' CoCo usually doesn't have enough system ram left to run the format command. This is actually pretty handy, since hard disks won't format either;-) but you do need a vanilla boot for system maintenance- the 'crash boot' is handy for this, and you'll use it often enough to make sure it works when the hd conks out.

Next, you'll need more window descriptors. MultiVue systems tend to use more windows than a shell driven system, so at least add the w8 through w15 descriptors included. More can be cobbled up by changing the internal names with DEd, or downloaded from many sources. Even if you've managed to get by without the generic /w descriptor in the past, you'll need it now! These are tiny anyway, so stuff 'em in.

The last step is to make MultiVue's utilities available. Some, like calendar, aren't really that handy and may be omitted. On floppies, you might want to load a few of these at boot to clear off the runtime /dd flop.

continued on page 15

Graphics Symmetry

OS-9 Level II Graphics Techniques

Robert Gault

The code presented in this article is written in C as used by the OS-9 C package from Tandy. Both because C is not my primary language and for speed considerations, I may have bent the rules of C style to create code
that is optimal at the assembly level. In fact, there are several sections of the graphics program that should be hand tuned at the assembly level given the inefficiencies of the Tandy C package.

I have indicated critical sections of code by making the C version a comment and including optimized asm code. You can easily do this for your own programs by having cc1 create a commented asm file for review.

There are times when speed is essential in programming. This is especially true when operating in real time. Shown below is a captivating graphics program which draws random patterns based on symmetrical reflections about the orthogonal and diagonal planes of a square. The program must plot thousands (5600) of points to create one pattern so speed is crucial.

To obtain maximum speed in plotting, the program can't afford and does not use the system overhead of going through the windowing commands. The screen is "poked" directly from the program. This would be a simple task under ROM Basic as the location of all graphic screens is well defined; not so in OS-9.

OS-9 Level I VR2.00.00 introduced hires Coco3 graphics screens which could be used from a VDG screen. The concept was retained in Level II. These special screens can be aquired via SetStat calls: SS.AScrn, SS.DScrn, SS.FScrn, and SS.PScrn. Happily SS.AScrn reports to the user the starting address of the screen.

My first version of Symmetry ran only on a VDG screen and found the address of the graphics screen via SS.AScrn. This did not satisfy me for there is no fool-proof method for a program to create a VDG screen and then switch to it. My program would have forced a user to initially create a VDG screen possibly requiring building a new boot file; not friendly!

There should be an analogous method of finding the start of a graphics window in Level II but there is not. The system must maintain a table of window addresses somewhere for there is an error code 193 "screen or window table is full". However, there is no reference to such a table in the
Def files on the direct page, in process tables, nor path descriptors. In short, the window table is hidden beyond my abilities to find it.

Luckily it is not necessary to make use of the "window table" for the direct page does contain images of the hardware bytes used for setting the vertical offset of the screen. If we can read D.VOFF1 and D.VOFF2, the data can be converted into the address of the start of the current window.

This technique works only for programs running on the currently active window. If a program changes the window type before reading D.VOFF1 & D.VOFF2, the progam must wait until the system can install the new window before the data will be valid.

"Symmetry" demonstrates how this technique works using the OS9 F$CpyMem call to read the offset values. The program also tests the current screen type and uses either VDG or window commands as apropriate. You can start the program from any screen: VDG, text or graphics window.

Warning! The patterns produced by Symmetry (with an RGB monitor for best results) are so facinatingly pretty that you could find yourself staring at the screen for hours. Well most Coco enthusiasts do that anyway so no harm done!

Editor: This woud make a great screen saver... in fact I have it running in a window of my CoCo 3 right now! Just load and run in a window, then flip to that window when you leave the computer... an excellent demonstration of CoCo capabilities...

```
/* Square symmetry: by Robert Gault Feb. 1994
   This program produces images based on reflection of a random point
about four axis; the two orthoginal and diagonal planes of a square.

   The code may seem odd to a full time C programmer. The intent is to
maximize speed through direct page variables and make hand tuning of
the intermediate asm file easier. Some optimization possible because
of known limits  to certain variables. */


#include <time.h>

#define I_SetStt 0x8e
#define I_GetStt 0x8d
#define F_MapBlk 0x4f
#define F_ClrBlk 0x50
#define F_CpyMem 0x1b
#define SS_AScrn 0x8b
     /* Assign and map hires graphics screen */
#define SS_DScrn 0x8c
     /* Display screen */
#define SS_FScrn 0x8d
     /* Release screen */
#define SS_ScTyp 0x93
     /* Report type of current screen */
#define SS_ScSiz 0x26
     /* Window size */
#define SS_FBRgs 0x96
     /* *  foreground, background, border report */
#define D_VOFF1  0x9d
     /* direct page image of $FF9D; vertical offset pixel values */
#define Hcenter 160
     /* horizontal center of screen */
#define Vcenter 95
     /* vertical   center of screen */

int interrupt();

extern char void;
point(void);
hpoint(void);

direct char rand[5];
     /*used by random number routine*/
direct char
*screen,*clrscr,*temp,color,keypress;
direct int i,n,sig,sequence,xx,yy,gx;
direct char gcolor,gmask;

     /* RGB color scheme */
char palette[ ]={0,1,8,3,2,16,6,34,38,
52,54,62,60,61,59,63};

      /* DISPLAY codes follow:
        set palette */
char pal_set[]={0x1b,0x31,0,0};

     /* kill current window & create a
      type8 */
char dwnew[ ]={0x1b,0x24,0x1b,0x20,
8,0,0,40,24,1,0,0};

     /* select window */
char dwselect[ ]={0x1b,0x21};

char window;
double x,y;
     /* used to maintain current graphic
      cursor location */

struct registers {
     char rg_cc,rg_a,rg_b,rg_dp;
     unsigned rg_x,rg_y,rg_u;
   };

struct registers reg;

main()
{
    int rnd(),*reg_d;
     unsigned getscreen(),offset;
    char ans;
     struct sgtbuf buffer; /* see time.h */
     intercept (interrupt);
     reg_d=&reg.rg_a; /* reg.rg_a & reg.rg_b
now addressed by reg_d */

      puts("\x0cWelcome to the world of");
      puts("squared symmetry.\n");
      puts("If you want instructions");
      puts("or information press \"y\"");
      puts("\n<hit any key to continue>");
      while (!read(0,&ans,1));
      /* pause until keypress */

     if(tolower(ans)=='y'){
       puts("\x0cThis program in C by");
       puts("Robert Gault is very simple");
        puts("to use; only two commands");
        puts("needed.\n");
        puts("<<shift>><<esc>> = halt");
        puts("  any other key restarts\n");
       puts("<<esc>>          = quit\n");
        puts("<hit any key to continue>");
```

```
    while (!read(0,&ans,1));
  }

  /* get random seed for rnd from
    system time */
  getime(&buffer);      /* see time.h */
   temp=&buffer;
  for(i=0;i<5;i++)
     rand[i]=temp[i+1];

  /* test screen type for VDG screen */
  reg.rg_a=1;
   reg.rg_b=SS_ScTyp;
   if(_os9(I_GetStt,&reg)==0)
  /*will return error only for VDG screen*/
  {
    /* true window so convert (if needed
     to type8) & find screen */
      if(8!=(window=reg.rg_a))
    {
        if(write(1,dwnew,12)==-1) exit(240);
       write(1,dwselect,2);
      tsleep(4);
/*give system time to setup new window*/
     }

     /*find address of window as vertical
     offset*/
     n=0;    /* MMU block */
     *reg_d=&n;
    /* point "reg.rg_d" to 'n' which contains
    MMU block # */
     reg.rg_x=D_VOFF1;
   /* offset in block to look at */
     reg.rg_y=2;    /* get two bytes */
     reg.rg_u=&offset;
    /* point reg_u to a storage buffer */
     if(_os9(F_CpyMem,&reg)==-1) {
       puts("Sorry can't find your window in
memory!");
      exit(0);
     }

    /* fgrnd=1 bgrnd=border=0 palette #'s*/
     system("display 1b 32 1 1b 33 0 1b 34
0 1b 31 0 3f 1b 31 1 0");

   /* map screen into our address space */
     reg.rg_b=4;
      reg.rg_x=(offset>>10)&0xff;
   /* convert vertical offset to MMU block */
      if(_os9(F_MapBlk,&reg)==-1) exit(207);
       screen=(Hcenter>>1)+Vcenter*160+
(clrscr=reg.rg_u);

     /* turn off cursor; etc. */
     system("display 5 20; tmode -echo");
   }
  else
  {    /* VDG type screen so forget
    windows */
     window=0;
   /* map high res screen into VDG
     space; initialize pointers */
      screen=(Hcenter>>1) + Vcenter*160 +
(clrscr=getscreen());

   /* display the graphics screen */
     reg.rg_b=SS_DScrn;
    reg.rg_y=1;
     _os9(I_SetStt,&reg);

   }

  /* set palette colors */
  for(i=0;i<16;i++){
    pal_set[2]=i;
     pal_set[3]=palette[i];
     write(1,pal_set,4);
   }

  /* setup for main infinite loop */
  for(;;){

  /* next routine is replaced by asm for
     speed clear graphics screen to palette
     0 which will be black

   for(n=0;n<0x7800;n++)
   clrscr[n]=0; */

  /* clear screen */
#asm
 ldu <clrscr * point reg.U to start of screen *
 leau 160*192,u * point to end of screen *
 pshs y * save direct page pointer *
 clra
 clrb
 tfr d,x * clear regs x & y *
 leay ,x
clr1 pshu d,x,y * clear 6 bytes *
 cmpu <clrscr * reached start of screen yet?
*
 bne clr1
 puls y * recover direct page pointer *
#endasm

     x=y=sequence=0;

    for(n=0;n<700;n++) {
     x+=(rnd(4)-1.8);
     if(x>90||x<-90)
      x/=(rnd(8)+2);
     y+=(rnd(4)-1.8);
     if(y>90||y<-90)
      y/=(rnd(8)+2);
     if(sequence==0){
       color=(color=(rnd(14)+1))<<4|color;
       sequence=rnd(6)+1;
     }
     --sequence;

     /* rather than passing parameters,
      external dp variables are forced */
     xx=x;
     yy=y;
     point();
     xx=y;
     yy=x;
     point();
    }
    sleep(10);
   /* give user a chance to view
      completed pattern */
  }
 }

/* stack checking can be removed here for
speed symmetry invoked is:
 +xx +yy
 +xx -yy
 -xx -yy
 -xx +yy
point()
 {
   hpoint();
   yy=-yy;
   hpoint();
   xx=-xx;
   hpoint();
   yy=-yy;
   hpoint();
  } */

#asm
 ttl point
point:
 lbsr hpoint
 ldd <yy
 nega
 negb
 sbca #0
 std <yy
 lbsr hpoint
 ldd <xx
 nega
 negb
 sbca #0
 std <xx
 lbsr hpoint
 ldd <yy
 nega
 negb
 sbca #0
 std <yy
 lbsr hpoint
 rts
#endasm

/* assign pixel mask; 2 pixels per byte plot
pixel on screen
hpoint()
 {
   if((xx&1)!=0)
     gmask=0xf0;
   else
     gmask=0x0f;

    temp=yy*160+(xx>>1)+screen;
    *temp=(*temp & gmask)|(~gmask &
color);
  } */

#asm
 ttl hpoint
hpoint:
 lda #$f0
 ldb <xx+1
 bitb #1
 bne oddpix
 lda #$0f
oddpix sta <gmask

 ldd <yy
 bpl posnum
 negb
posnum lda #160
 mul
 tst <yy
 bpl posnm2
 nega
 negb
 sbca #0
posnm2 pshs d
 ldd <xx
 asrb
 addd ,s++
 addd <screen
 std <temp

 ldb [>temp,y]
 andb <gmask
 pshs b
 ldb <gmask
 comb
 andb <color
 orb ,s+
 stb [>temp,y]
 rts
#endasm
```

continued on page 17

The Hardware Hacker

External SCSI cable for the MM/1

Chris Hawks

Editor: The following describes how to make an adapter cable from a standard 50 SCSI connector to the 25 pin variety normally found on a Macintosh and some portable PC type SCSI CD-ROM drives.

I have had a few requests regarding the nature of SCSI cables for the MM/1 and external CD-ROM Drives. Here is the scoop on how I made mine:

Three items are required. You need a 50 pin IDC plug (connects to the SCSI connectors on your existing SCSI cable), a DB25 female connector (connects to the DB25 male on the cable supplied w/ CD-ROM), and 2 feet of 50 conductor ribbon cable.

First, crimp the 50 pin IDC plug to one end of the ribbon cable. Match pin #1 of the cable (usually a red stripe) to pin #1 of the connector (usually has a small arrow-head pointing to it).

Second, at the other end of the ribbon cable, separate all 50 wires of the cable for about two inches.

Third, cut off all of the odd numbered wires from the ribbon cable (be careful!).

Fourth, solder (or crimp, depending on the type DB25 you have) the wires of the ribbon cable to the DB25 as follows.

Ribbon Cable end	DB25	Signal
2	8	data bit 0
4	21	data bit 1
6	22	data bit 2
8	10	data bit 3
10	23	data bit 4
12	11	data bit 5
14	12	data bit 6
16	13	data bit 7
18	20	data parity
20	7	Gnd
22	9	Gnd
24	24	Gnd
26	25	terminator pwr
28	18	Gnd
30	14	Gnd
32	17	attention
34	16	Gnd
36	6	busy
38	5	acknowledge
40	4	reset
42	2	message
44	19	select
46	15	command/data
48	1	request
50	3	I/O

The ground connections may be interchanged if you like.

Last, install the DB25 connector in a cut-out on the back of your MM/1 case and plug the 50 pin connector into your SCSI cable. Then the external SCSI cable can be used to connect your CDRom!

< 268'm >

Operating System Nine

continued from page 12

The Media Challenged....

And here I'm going to break with tradition- if you have a 360K and a 720K drive, make the 720 /dd. You don't have to make it /d0, and you don't have to make a 720K boot disk. Instead, copy MODULES/d1_80d.dd to ddd1_80d.dd, and use 'dEd' to rename it internally. Inside, you'll see it's name 'd1' on the ascii side, but if you look at the hex numbers, the '1' isn't $31 like you'd expect, but $B1. OS9 marks the end of a string like this, setting the high bit. Use hex edit to change 44 B1 (d1) to 44 C4 (dd), verify/save, and Gen a boot with this ddd1_80d.dd module as /dd. This is not a normal boot disk, since most everything (cmds, sys (except stdpats etc, which you can explicitly merge from /d0 in startup), has to be on the 720K drive, not the boot disk.

Why? The boot floppy /d0 only has to hold OS9 boot- less than 50K. Startup (which is on /dd) can then explicitly load the std* files and your most used programs from /d0 (load /d0/stdfonts /d0/stdpats /d0/textwhacker.....) Note the single load line lumps all the programs into a same data block- big data programs need a separate load command, so they don't carry any extra code to their 64K space. Archane example- 'load ds df' doesn't work very well, while 'load ds; load df' is fine, each gets a separate dataspace.

The /d1 nee /dd disk should contain all commands that aren't preloaded and aren't very big- your utilities and handy stuff like gsort. And _all_ the icons. It should also contain all your shellscripts (also in /dd/cmds) and all the sys, lib, defs and etc. a /dd/usually has. This may force you to use a 'user' disk and a 'author' disk, but basically the result is the 'one true /dd' that MultiVue likes to work under. The data disks that are swapped into /d0 only have to be clicked on, no 'change execute'. They can also contain programs, which have to loaded using the AIF calls a shellscript system. Plus, if you stick in a Tandy bootgame, it works OK from the original format disk, which is still /d0 and doesn't know you moved /dd.

With dual 360's, you can't avoid 'change execute'. The stratigy is to boot and load what you can from startup, then 'change execute' to a different disk in /d0. This 'different disk' is geared more towards the 'dd stuff'- it should have all the icons and SYS files, plus unloaded cmds (you should have most of the original cmds in two preloaded startup files by now). Don't worry about programs- just make sure all the hardcoded stuff like c compiler libraries are on /dd.

Now, get to writing script files. Remember the extended 'run a script or B09 idea'? You want your AIF to call a script (probably, instead a B09/runB which might run the dang script anyway- to fix the icon name bug) that first loads the program from /d1/cmds, then runs it from the current data dir (and unlinks it later, if you want to get sweaty). To use the built in program/data.ext link, you'll need a second load icon- name it AIF.AAA or something- make a nice bomb ICON for it;-) This loader AIF doesn't have to just load one program- with enough RAM, bomb could load every app used by the data on the disk.

I know, this sounds really strange. But I've tried (and failed) to get the 'normal' way going. It's just too hard to keep the cmds with the icons and the aifs and the data on floppie disks. Add 'disk:change execute+type_stuff' and ftftppthhptpth ACK!

And that about completes MultiVue Two, except for the stuff I left out.

< 268'm >

**** PRESS RELEASE ****

The coding for Digigrade Productions' new adventure software is completed. Full product expected in late Feb. '96.

Features include:

Half a city block to explore!
Events change your surroundings!
Myst type look and feel!
Frightening locations digitized in 256 colors!
Mysteries to solve!

This is it my fellow OS-9ers! The MM/1 adventure you have been waiting for! After months of thought, and weeks of programming, I have developed an adventure driver and files that let you switch directories and play a totally new adventure! There is no stopping a release of 2 - 3 adventures using this software!

Keep your eyes glued to these pages for more info OR send a postcard to:

Digigrade Productions
C/O Dave Pellerito
1325 West Lincoln Hwy. #412A
De Kalb, IL. 60115

GLENSIDE COLOR COMPUTER CLUB

PROUDLY ANNOUNCES THE FIFTH ANNUAL

"LAST" CHICAGO CoCoFEST!

WHERE:

Elgin Holiday Inn
(A Holidome Family Recreation Center)
Special 'Fest Rate: $57 per night
Call 708-695-5000 for reservations
A limited number of rooms will be held for the fest!
They will not be held after 28 March.

WHEN:

April 13 (10-5)
April 14 (10-3)

PRICE:

Non-members: $10
Glenside Members: $5
(Glenside memberships: $15/year)
Please send an SASE or add $0.50
for postage and handling per order.
Vendors: Inquire for booth prices.

Ticket and Membership Sales:
George Schneeweis, Treasurer
Glenside Color Computer Club
RR#2 Box 67
Forrest, IL 61741-9629

Further Information:
Eddie Kuns; 708-820-3943 voice
eddiekuns@delphi.com e-mail
Tony Podraza; 708-428-3576 voice
708-428-0436 BBS (8-N-1, 2400 baud)

FARNA Systems

Software, Books, and Hardware for all OS-9/OSK Systems!
ADD $2.50 S&H
($4.00 Canada, $10.00 Overseas)

Box 321
Warner Robins, GA 31099
Phone 912-328-7859
Internet: dsrtfox@ delphi.com

CoCo DECB Software:

CoCo Family Recorder - $12.00
Genealogy program for CoCo 3. Requires 2 drives, 80 col. monitor.
OS-9 Version - $20.00 (details below)

DigiTech Pro - $12.00
Record any sound for easy play-back in your BASIC or M/L programs. CoCo 3 512K.

ADOS: Support for double sided drives, 40/80 tracks, faster formatting, much more!
Original (CoCo 1/2) - $15.00
ADOS 3 (CoCo 3) - $25.00
Extended ADOS 3 - $30.00 (ADOS 3 req., RAM drives, support for 512K-2MB)
ADOS 3/Ext. Combo - $50.00

CoCo OS-9 Software:

Patch OS-9 - $7.50
Automated program installs most popular/ needed patches for OS-9 Level II. 512K and two 40T/DS (or larger) drives required. (*128K /35T users can install manually- state 35T.*)

CoCo Family Recorder/OS-9 - $20.00
Same features as DECB version. Data can be transferred from DECB version! Will run on a single 720K drive, advanced editing windows.

Pixel Blaster - $12.00
High speed graphics tools for OS-9 Level II. Easily speed up your game programming with this tool kit and subroutines!

OS-9 Point of Sale - $62.50
Maintain inventory, print invoices, customer catalog, etc. Multi-user capable . Supports ASCII terminals. Menu driven. CoCo3.

Books:

NEW!! **Mastering OS-9 - $30.00**
This is the long awaited update of Paul Ward's "Start OS-9". New format is easier to read, has easier to follow tutorials,and updated information files. Comes complete with disk, which has a few added utilities.

Tandy's Little Wonder - $22.00
History,technical info, schematics, peripherals, upgrades, modifications, repairs, much more- all described in detail for all CoCo models! Vendors, clubs, BBSs also listed.

Quick Reference Guides
OS-9 Level II - $5.00 OS-9/68K - $8.00
These handy QRGs have the most needed info in a 5.5"x 8.5" desk-top size. Command syntax, error codes, special keys functions, etc.

FARNA Systems Publishing Services

Type Setting and Printing: We can prepare professional typeset manuals, books, booklets, catalogs, and sales flyers for you - we can print or you reproduce as needed from a master set! Very reasonable prices - inquire!

Contact Frank Swygert at FARNA Systems for quotes

CoCo Hardware:

DigiScan Video Digitizer - $150: Capture images from VCR, camcorder, or TV camera. No MPI required- uses joystick ports. CoCo Max3, Max 10, Color Max 3 compatible. Special order- allow 90 days for delivery. Send $75 deposit, remainder due before delivery.

Puppo Keyboard Adapters - $70: Use IBM PC/XT keyboards with your CoCo! Mounts in your CoCo case with no soldering. 101 key keyboards available for $30 with order.

listing continued from page 15

```
interrupt(sig)
int sig;
{
  char ans;
  /* shift break */
  if(sig==3){
     while (!read(0,&ans,1)||ans==3);
    /* pause until another keypress */
    return;    /* go back to calling routine */
   }
   if(window==0) {
     /* VDG screen so .... */
     reg.rg_a=1;
      reg.rg_b=SS_DScrn;
     reg.rg_y=0;       /* flag text screen */
       _os9(I_SetStt,&reg);
     /* display indicated screen */
     system("display 1b 30 c");
     /* default colors, clear screen */
     for(n=1;n<4;++n){
       reg.rg_b=SS_FScrn;
       reg.rg_y=n; /* graphics screen #1

*/      if(!_os9(I_SetStt,&reg)) break;
     /* release screen from memory */
      }
    }
   else
     {      /* true window so .... */
       reg.rg_b=4;
      /* number of blocks to release */
       reg.rg_u=clrscr;
     /* point to start of first block */
       _os9(F_ClrBlk,&reg);
        system("tmode echo; display 5 21");
     /* cursor on */
       if(window!=8)
     {
         write(1,dwnew,2);
      /* device window end */
      /* os-9 will restart original window */
       }
      else
        system("display 1b 30 c");
     /* set default colors */
     }

  if(sig==2)
   /* report signal to caller unless BREAK */
   sig=0;

  exit(sig);
 }

unsigned getscreen()
{
   reg.rg_a=1;
   reg.rg_b=SS_AScrn;
    /* allocate and map screen into address
      space */
   reg.rg_x=4;
     /* 320x192 16 colors 16K screen */
   if(_os9(I_SetStt,&reg)==-1)
    /* if call does not work */
     interrupt(reg.rg_b);
    /* return error to user */
   return(reg.rg_x);
     /* return address of screen */
 }

     /* Stack checking can be removed here
       for speed increase */
int rnd(val)
       /* standard rnd rountine replaced with
          faster asm; no floats */
unsigned val;
{
#asm
 ldb 5,s
 incb
 andcc #$fe
 lda <rand+4
 bita #1
 beq r1
 lda #8
 eora <rand
 sta <rand
 lda #128
 eora <rand
 sta <rand
 orcc #1
r1 ror <rand
 ror <rand+1
 ror <rand+2
 ror <rand+3
 ror <rand+4
 pshs b
 lda <rand+1
 mul
 ldb ,s
 sta ,s
 lda <rand
 mul
 addb ,s+
 adca #0
 clrb
 stb 4,s
 sta 5,s
#endasm
 return val; /* whole number returned; no
fractional part */
 }
```

< 268'm >

Wittman Computer Products

7376 North Bergen Road
Bergen, NY 14428
Phone 716-494-1506 : Fax 716-293-1207
Internet: ww2150@acsprl.acs.brockport.edu

NEW ADDRESS!

K-Windows Chess for MM/1
Play chess on your MM/1.....$24.95

X-10 Master Control for MM/1
Use MM/1 to control you home!.....$29.95

Variations of Solitaire
Pyramid, Klondike, Spider, Poker and Canfield
MM/1.....$29.95 CoCo3.....$19.95

OS-9 Game Pack
Othello, Yahtzee, KnightsBridge, Minefield, and Battleship
MM/1.....$29.95 CoCo3.....$19.95

WPShell
An OS-9 Word Processing Point and Click Interface
CoCo3.....$14.95

Using AWK with OS-9
Includes V2.1.14 of GNU AWK for OS-9/68000
MM/1.....$14.95

NEW! WCP306 Computer!
MC68306 16.67MHz CPU, code compatible with MC68000. Four SIMM sockets, 512K-16MB memory. IDE hard disk interface, floppy interface, two serial ports, parallel port, real-time clock, all built into motherboard! Designed to use 16 bit AT expansion cards (six slots) and standard AT keyboard, power supply, and case. Comes with "Personal OS-9/68000" and MGR, a graphical user interface with complete documentation. Only $400 for bare board as described above!

Call or write for a free catalog! Demo disks also available.
Owned and operated by William L. Wittman, Jr.

New products!

Basicboost: 6309 port of Basic09's RunB module. Packed programs run up to 15% faster (varies w/ functions used).
Screenboost: 6309 version of the CoCo's screen drivers. Noticeably speeds up most screen functions. Adds support for screens up to 200 lines for gfx and 28 lines and 128 columns for text windows plus horizontal scrolling. New commands to manipulate gfx fonts and one that allows programs to move and/or resize window in which they run.
$10 each or $16 for the pair.

HSlink: Null-modem file transfers with other computers (CoCo or not). Uses printer port - no special hardware needed. ASCII and Xmodem transfers up to 19200 baud (6809) and 57600 baud (6309). Cables can be made on request (specify CoCo, DB-9, or DB-25 cable).
$14.95 or $24.95 with cable.

Prices in US Dollars. Send for Canadian dollar prices.
Add $3 US, $5 Canada for shipping.

C. Dekker ...

RR #4 Centreville, NB
E0J 1H0, CANADA
Phone 506-276-4841

User-friendly Level II Programs!

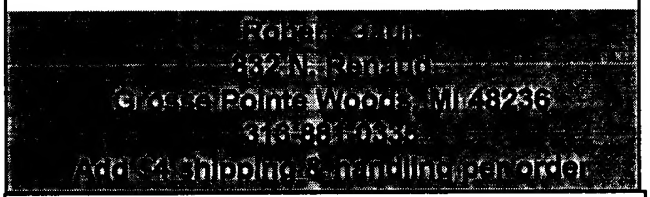
EDTASM6309 Version 2.02 ---------- $35.00
This is a major patch to Tandy's Disk EDTASM to support Hitachi 6309 codes. Supports all CoCo models. CoCo 3 version uses 80 column screen, 2MHz. YOU MUST ALREADY OWN TANDY'S DISK EDTASM TO MAKE USE OF THIS PRODUCT. It WILL NOT work with a disk patched cartridge EDTASM.

CC3FAX ---------- $35.00
Extensive modification to WEFAX (Rainbow, 1985) for 512K CoCo 3. Uses hi-res graphics, holds full 15 min. weather fax image in memory. Large selection of printer drivers. Requires shortwave receiver and cassette cable (described in documentation)

HRSDOS ---------- $25.00
Move programs and data between DECB and OS-9 disks. Supports RGB-DOS for split DECB/OS-9 hard drives. No modifications to system modules (CC3Disk or HDisk) required.

DECB SmartWatch Drivers ---------- $20.00
Access your SmartWatch from DECB! New function added to access date/time from BASIC (DATE$). Only $15.00 with any other purchase!

RGBOOST ---------- $15.00
Make the most of your HD6309 under DECB! Uses new 6309 functions for a small gain in speed. Compatible with all programs tested to date! Only $10.00 with any other purchase!



for all your CoCo hardware needs, connect with

CoNect

449 South 90th Street
Milwaukee, WI 53214
E-mail:pulland@omnifest.uwm.edu

The main problem with OS9 under a CoCo is the serial port. With a one character buffer, it's hard to do much before the serial port needs service. Windows and OS2 have the same problem. Or did, until National released the 16550 uart- 16 bytes of internal fifo buffering gives multitasking systems time to do some.

Announcing Fast232

Tandy with Sacia			**Fast232**	
bps	load	thruput	load	thruput
2400	28.3 sec	237 cps	25.3 sec	235 cps
9600	73.6 sec	938 cps	31.4 sec	950 cps
57600	not available		32.6 sec	5373cps

Local machines withfaster modems can now be connected to properly (or improperly at 115K!). More down to earth, Fast232 is a ROMPak sized case, which will accept a daughterboard (once I get the case to close) to give two ports in a very 'concise' package. OS9 drivers by Randy Wilson. Free bonus software! The pd release of 'SuperComm' (Dave Phillipsen and Randy Wilson). All software includes 6809 and 6309 versions.

Fast232 $79.95
Second port $45.00

The OS-9 User's Group, Inc.

Working to support OS-9 Users

Membership includes the Users Group newsletter, MOTD, with regular columns from the President, News and Rumors, and "Straight from the Horse's Mouth", about the use of OS-9 in Industrial, Scientific and Educational institutions.

Annual Membership Dues:

United States and Canada	Other Countries
25.00 US	30.00 US

The OS-9 Users Group, Inc.
6158 W. 63d St. Suite 109
Chicago, IL 60638
USA

Northern Xposure

'Quality Products from North of the Border'

OS-9 Level II Color Computer 3 Software

NitrOS-9 v1.20 Call or write for upgrade info or new purchase procedure. Requires Hitachi 6309 CPU	$29.95
Shanghai:OS-9 Introductory price Send manual or RomPak to prove ownership	$25.00
Thexder:OS-9 Send manual or RomPak to prove ownership	$29.95
Smash! Breakout-style arcade game	$29.95
Rusty Launch DECB/ECB programs from OS-9!	$20.00
Matt Thompsons SCSI System v2.2 'It flies!' 256/512 byte sectors, multipak support	$25.00

Disk Basic Software

Color Schematic Designer v3.0 New lower price	$30.00
Oblique Triad Software	*Write for catalogue*

Color Computer 3 Hardware

Hitachi 6309 CPU (normally 'C' model, may be 'B')	$15.00
SIMM Memory Upgrade Runs Cooler! 512k $44.95	0K $39.95
Sound Digitizing cable	$15.00

OS-9/68000 Software

OSTerm 68K v2.2 External transfer protocol support TTY/ANSI/VT100/K-Windows/Binary Emulation	$50.00
Upgrade from TasCOM (Send TasCOM manual please)	$30.00

7 Greenboro Cres
Ottawa, ON K1T 1W6
CANADA
(613)736-0329

All prices in U.S. funds.
Check or MO only.
Prices include S&H

Internet mail: cmckay@northx.isis.org

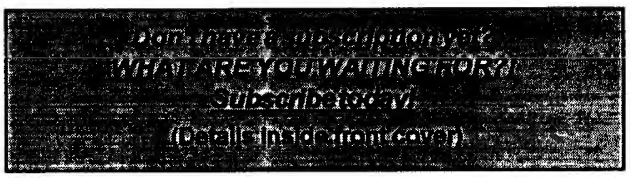
Have you been wishing you had subscribed to microdisk? Can't justify the cost? Well, prices have come down! Now what's your excuse?

New "microdisk" subscription rates:

US and Mexico:	$20/year
Canada:	$25/year
Overseas (airmail):	$40/year
Single issues:	
US and Mexico:	$8/each
Canada:	$10/each
Overseas (airmail):	$15/each

One issue of microdisk is published for every two issues of the world of 68' micros (six magazines, three disks per year)

ADVERTISER'S INDEX:



For superior OS-9 performance, the

SYSTEM V

Provides a 68020 running at 25 MHz, up to 128 MBytes of 0 wait-state memory, SCSI and IDE interfaces, 4 serial and 2 parallel ports, 5 16-bit and 2 8-bit ISA slots and much more. The **SYSTEM V** builds on the design concepts proven in the **SYSTEM IV** providing maximum flexibility and inexpensive expandability. Optionally available at 33 MHz.

\N OS-9 FIRST - the MICROPROCESSOR is mounted on a daughter board which plugs onto the motherboard. This will permit inexpensive upgrades in the future when even greater performance is required.

G-WINDOWS benchmark performance index of a **SYSTEM V** running at 25 MHZ with a standard VGA board is 0.15 seconds faster than a 68030 running at 30 MHz with an ACRTC video board (85.90 seconds vs 86.05 seconds).

For less demanding requirements, the

SYSTEM IV

The perfect, low cost, high-quality and high performance OS-9 computer serving customers world-wide. Designed for and accepted by industry. Ideal low-cost work-station, development platform or just plain fun machine. Powerful, flexible and expandable inexpensively. Uses a 68000 microprocessor running at 16 MHz.

G-WINDOWS - a PROVEN WINNER

FOR OS-9

Available for the **SYSTEM IV** and **SYSTEM V** computers, the PT68K4 board from Peripheral Technology and the CD68X20 board from Computer Design Services. Resolutions from 640 x 480 x 256 to 1024 x 768 x 256. Support for multiple VGA cards running different processes, different portions of the same process or both.

G-WINDOWS is now AVAILABLE for OS-9000 from *delmar co*

Support for generic VGA boards and SUPER VGA boards including ET4000 (Tseng Labs), OTI067 (Oak), CT452 and CT453 (Chips and Technology), GENOA, WD90C11 (Paradise) and S3 (S3 Inc.) for modes from 640 x 480 to 1280 x 1024 depending on board.

Distributor of MICROWARE SYSTEMS CORPORATION Software

This ad was prepared and printed using QuickEd under OS-9.

delmar co

PO Box 78 - 5238 Summit Bridge Road - Middletown, DE 19709

302-378-2555 FAX 302-378-2556